POLITECNICO DI MILANO
Department of Electronics, Information and Bioengineering
Master of Science in Biomedical Engineering

Master Thesis

GUESS MY GESTURE:

A gesture recognition algorithm in a robot therapy for ASD children.

***Supervisor*:**
Prof. Alessandra Pedrocchi

***Co-Supervisors*:**
Eng. Laura Santos
PhD Alice Geminiani

Master Thesis of:
Giubergia Alice, ID 920605
Ivani Alessia Silvia, ID 914185

Academic Year 2019-2020

Abstract

Introduction
As reported by the World Health Organization (WHO), Autism Spectrum Disorder (ASD) is a neuro-developmental condition characterized by some degree of impaired social behaviour, communication and language and a narrow range of interests and activities that are both unique to the individual and carried out repetitively. It is estimated that worldwide 1 in 160 children has ASD [1]. Intervention during early childhood is crucial to promote development and long-term positive effects. Children with ASD may have difficulty communicating non-verbally, such as through hand gestures, eye contact, and facial expressions [2].

Gold-standard therapy centres on the delivery of evidence-based psycho-social interventions, trying to adapt to the needs of each child. Recent studies have shown that children with autism cope well with rule based, predictable systems such as humanoid robots. ASD children feel more comfortable with such robots than in the presence of humans, who may be perceived as hard to understand and sometimes even frightening [3]. The "supervised autonomy" in which the humanoid robot works independently under a supervisor's guidance is known as Robot-Enhanced Therapy (RET) [4].In this way, the Therapist-Robot-Child triadic interaction is elicited, facilitating communication between child and therapist. As observed by Sial et al. in their work [5], a collaborative approach based on interactive games between a robot and ASD children has produced positive results in terms of therapeutic outcomes such as social interaction, communication, joint attention and turn taking. Studies have shown that motor impairments are a prominent comorbidity within the ASD phenotype [6], even though they are not currently included in the diagnostic criteria of autism. McAuliffe et al. [7] explored how these altered motor ability might be linked to an atypical skills learning. In point of that, their results supports the hypothesis that a poor imitative gestural learning can impact social and motor development, since learning via imitation is a prime method by which humans acquire skills. As a consequence, teaching gesture imitation in RETs might improve the child's social skills as well as spontaneous gesture use [2]. Designing RET protocols with increased robot's autonomy is important in order to decrease the human workload and to deliver consistent therapies. In this context, gesture recognition algorithms can be exploited to trigger robot's feedback in interactive protocols and to evaluate children's performances. A proper feedback should be triggered to increase children's engagement, thus empowering the therapy's robustness.

Human action recognition relies on capturing systems able to track body's movements. Since children with autism are particularly sensible to the touch [8], wearable systems would be unfeasible, even if they are fast, robust and receive information directly from

users' movements . As a consequence, many approaches have exploited RGB, depth, or a combination of these two data types (RGB-D) from video sequences to recognize actions. However, visual-based capturing systems make the recognition a challenging task due to many factors such as occlusions, viewpoint, lighting and user-variance. Among those, low-cost depth cameras such as Microsoft Kinect v2 are able to provide a powerful human body tracking in real-time [9]. Thus, Kinect skeleton data (3D coordinates of body joints) can be used to distinguish many actions.

Deep learning-based approaches achieved promising results in classification tasks [10]. In the context of action recognition, Convolutional Neural Network (CNN)s are the most used. Particularly, Residual Network (ResNet)s, which are based on the learning of error functions (i.e. residual functions), are a good solution to extract with precision relevant features from biomechanical data, allowing a fast training process and resolving the vanishing gradient problem.

Since networks learn to recognize from data they are fed with, training datasets affect their behavior. The most used are public datasets and rarely newly created. Their gesture sets are usually selected from Activities of Daily Living (ADL) (tasks of every-day life) or from context-specific activities (such as the Gaming 3D Dataset [11]). Body parts involved in the actions range from a single limb, just the upper/lower body part or the whole skeleton.

The main goal of this thesis project is to find a proper method to automatically recognize gestures inside a robot therapy (IOGIOCO) for children with ASD. In this way, the robot could react properly to children's movements when interacting with them and support therapist's work in the Therapist-Robot-Child triadic interaction. The final goal is to implement and test an online algorithm in a clinical application to be able to help therapists, empowering children's learning.

Methods

IOGIOCO robot therapy is based on interactive mirroring games between the humanoid robot NAO and ASD children. It includes 5 training levels and one final evaluation and involves the training of selected communicative gestures from ADL. The 19 gestures of the protocol are: *tall*, *angry*, *listening*, *waiting*, *kissing*, *short*, *giving*, *where*, *hungry*, *me*, *peekaboo*, *happy*, *yes*, *no*, *big*, *hello*, *little*, *pointing*, *coming* (Figure 1). The recognition algorithm is involved from Level 3 on.

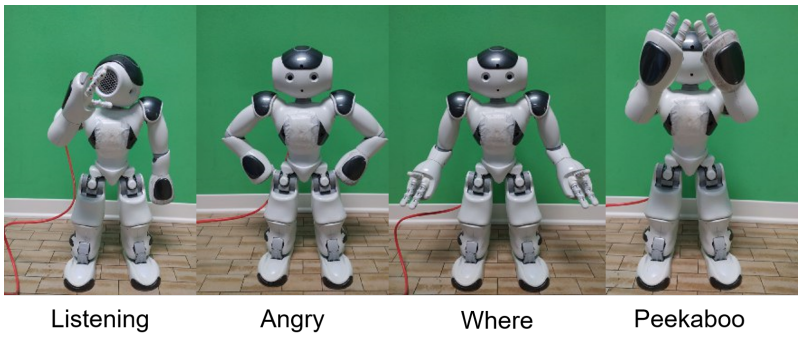

Figure 1: Some gestures perfromed by NAO: listening, angry, where, peekaboo and kissing.

The gesture recognition algorithm's workflow involved data acquisition, data processing and generation of pose features to facilitate the learning process of the classifier, the classification with Neural Networks and the online implementation (Figure 2).

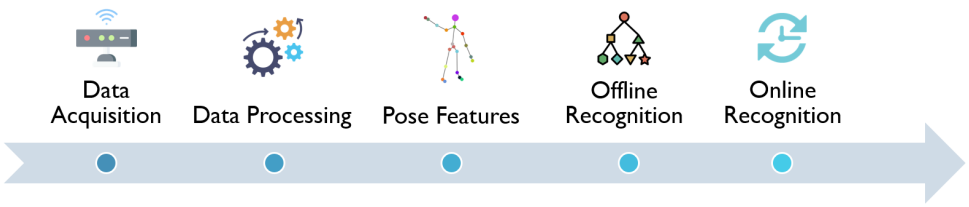

Figure 2: Workflow of the proposed algorithm.

Data Acquisition

The proposed method exploits Kinect camera to acquire data since a non-intrusive vision-based capturing system is required to monitor children and therapist's movements. Kinect camera is able to capture keypoints (skeleton 3D joints' coordinates) allowing a spatial analysis of each gesture.

Three different datasets were acquired and used for the development of the algorithm of gesture recognition. Among these, the last 2 were manually segmented. They were:

1. Subsampled Healthy Dataset: 5-gestures dataset with *small*, *hello*, *pointing*, *come* and *yes* gestures. 18 healthy adult subjects performed these actions once.

2. Healthy Dataset: 14-gestures dataset made of 18 healthy subjects, 9 adults and 9 children. The gesture set included *tall*, *angry*, *listening*, *waiting*, *kissing*, *short*, *hello*, *giving*, *where*, *hungry*, *big*, *little*, *pointing* and *coming* gestures.

3. Expanded Dataset: 19-gestures dataset made of 22 subjects, of which 2 were adults with ASD. Only 11 subjects of this dataset performed every type of action. The complete gesture set included *tall*, *angry*, *listening*, *waiting*, *kissing*, *short*, *giving*, *where*, *hi*, *peekaboo*, *hungry*, *happy*, *big*, *me*, *no*, *little*, *pointing*, *yes* and *coming* gestures.

Data Processing

Information extracted from Kinect was processed. Filtering was necessary to reduce noise by which Kinect is affected. Since coordinates were referred to the camera, the reference system was centered with respect to the subject himself. Different reference points to achieve *translation invariance* were tested. Furthermore, since healthy and ASD adults and children with different physical structures were involved in this project, normalization was required to obtain *user-invariance*. A frame by frame normalization was carried out. This was because of Kinect's low spatial resolution on depth data [12] and subject's undefined position in front of the camera, which could affect the normalization's value thus not reflecting the real body size. This approach led to an increasing in the robustness of the normalization's method. However, a frame by frame normalization changed the length of the scaling segment, computed as Euclidean distance, from one frame to the adjacent ones: this is true especially when dealing with, for example, *yes* gesture or any gestures involving a forward bend of the torso. In order to deeply investigate how much these

segments changed from one frame to another, for each gesture class the average segment's length over frames in every sample was computed. Then a comparison between segments' variations during the action for each gesture class was achieved by means of Standard Deviation.

Furthermore, Kinect data needed to be arranged according to biomechanics in order to highlight the kinematics characterizing a particular gesture in a single sample. Since the lower body was not crucial in the gesture set, only the upper body segments were preserved. To get an effective representation of the gesture executed and to keep local motion characteristics, joints' coordinates of upper body skeleton sequences were grouped into body sets (two arms and one trunk). Body sets were organized from top to bottom according to the physical structure of the human body (head and trunk first, right arm and left arm then). rearranged according to the physical structure of the human body . Thus, 3D matrices describing actions were generated by stacking together every frame of the movement, composed by 3D arrays of joints' coordinates.

Pose Features

After data processing, a focus on the meaning of network's inputs was carried out by generating *pose features* able to describe the kinematics of actions. The proper preparation of these features could somehow drive the learning process of the classifier [13]. To obtain a *pose feature*, all the 3D coordinates (x_k, y_k, z_k) of each frame F_t in a skeleton sequence were scaled through a normalization function $\mathbf{N}(\cdot)$:

$$\begin{aligned}(x'_k, y'_k, z'_k) &= \mathbf{N}(x_k, y_k, z_k)\\ x'_k &= \frac{(x_k - x_{min})}{(x_{max} - x_{min})},\\ y'_k &= \frac{(y_k - y_{min})}{(y_{max} - y_{min})},\\ z'_k &= \frac{(z_k - z_{min})}{(z_{max} - z_{min})},\end{aligned} \tag{1}$$

where (x'_k, y'_k, z'_k) are the normalized coordinates of k-th keypoint and $\mathbf{c_{max}}(x_{max}, y_{max}, z_{max})$ and $\mathbf{c_{min}}(x_{min}, y_{min}, z_{min})$ are the scaling coordinates. In order to standardize every action class, different *gesture normalization*s' techniques were experimented. In the end, keypoints were scaled exploiting a *gesture independent* normalization. Therefore, maximum and minimum coordinates of each channel (x, y, z) of every movements' sequence were detected whatever gesture executed and used as scaling values. Moreover, $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ were selected independently of body joints, thus in a whole-body control volume. These scaling coordinates could have been computed with respect to the entire dataset, thus obtaining different normalization values for each action in the gesture set. However, this approach would have been dataset specific, while a *gesture independent* normalization made the system *dataset independent*.

When using Artificial Neural Network (ANN)s and the only available motion features are skeletal data, an intermediate representation of skeletal sequences can help in data processing and in understanding samples the net has to learn from. Therefore, RGB pose features were obtained by transforming the coordinate space into RGB color space scaling each coordinate in the range of [0, 255]. In this way, kinematics of each action was preserved and outlined by a new image representation. During a movement, a displacement

in the x direction is depicted by a variation of red amount, while a shift in y or z direction corresponds to a change in green or blue, respectively. Figure 3 shows the way a pose feature representation is generated.

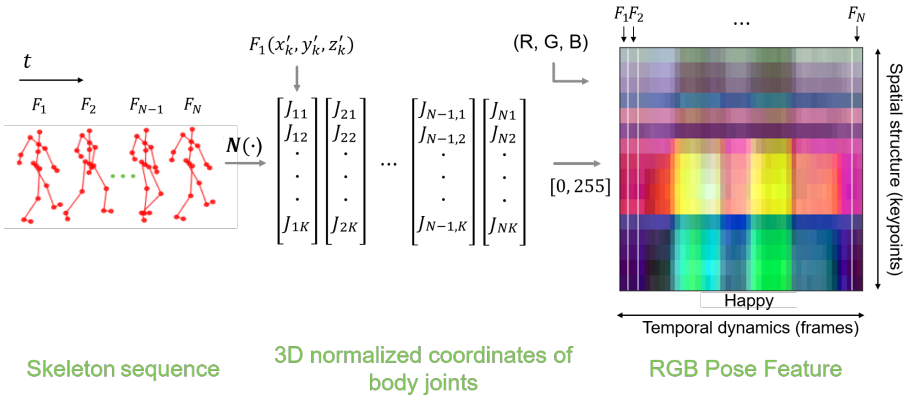

Figure 3: From skeleton sequences to RGB pose features: Every frame F_t of a sequence is transformed in a 3D array to be stacked in a matrix together with the others. N denotes the number of frames in each sequence and K denotes the number of keypoints in each frame. Then, after the gesture normalization, a scaling in the RGB color space is applied, thus obtaining a single RGB pose feature representing the motion. On the horizontal dimension temporal dynamics is shown, while the spatial structure (keypoints) is depicted on the vertical one.

Starting from this skeleton-based representation, further data processing was applied:

- Temporal Interpolation and Reshape: since CNNs require inputs to be the same length, different ways of interpolation and reshape were tested;
- Enhanced Action Images: a local contrast enhancement technique was exploited to further highlight the characteristics of the motion;
- Data Mirroring: since gestures used in the training of the model were executed with the right arm, data mirroring was able to make the classifier independent of the hand used to execute the action and to augment the dataset.

Classification

A Residual Network (ResNet) was implemented since is able to build a deep neural network without the risk of degradation in performance. This is possible thanks to skip connections, which allow the net to skip the training of one or more weight layers. Moreover, a Softmax function is frequently used in the last layer of the network. Softmax turns the numeric output of the last linear layer of a multi-class classification neural network into a vector of N probabilities, where N is the number of classes. So ResNet's output was a vector representing the probability distribution of all the 19 potential outcome gestures. *Pose features* were used as net's inputs. In order to evaluate the recognition algorithm, a Leave-P-Out subject cross-validation method was exploited for the most comprehensive 19-gestures Expanded Dataset. In this way, P out of N subjects in the dataset were used for testing and P−N for training and validating the model (P=2 subjects and N=11 subjects of the Expanded Dataset). Different net's hyperparameters were tested to achieve the best recognition's results possible.

Online Recognition

Once the algorithm was established offline, an online implementation was designed. It included two steps:

- Kinect-only configuration: the model was set and tested on the continuous data stream captured by the camera;
- Kinect-NAO configuration: the model was set and tested with the robot.

Different settings were experimented to take into account Kinect's behaviour with robot connection. For this reason, Kinect camera's Frames Per Second (FPS) was evaluated in both configurations to monitor the frame frequency, since the recognition task relied on data acquisition of skeleton poses.

In order to exploit the recognition algorithm in a real-time classification, a *sliding window* was used. Pose features were computed and analyzed by the classifier on a certain window, characterized by two configuration parameters: *size* and *step*. A window of fixed *size* in terms of number of frames was used on the continuous data stream captured by the camera. Different configurations were experimented to avoid lag between the performance of the action and the classification's output as much as possible. To detect the presence of a gesture among *no gestures*, the highest conditional probability output by the Softmax layer of the classifier was compared to a threshold $\tau \in [0,1]$:

$$state = \begin{cases} gesture, & \text{if } probability > \tau \\ no-gesture, & \text{if } probability < \tau \end{cases} \tag{2}$$

When the detection threshold was exceeded, the probabilities' prediction vector was saved in a *buffer*. Then, the window slided of a fixed *step* before predicting again (Figure 4). Once the *buffer* was filled with N prediction vectors, the algorithm identified the gesture performed with one of the following two possible methods:

- By averaging *buffer*'s prediction vectors' probabilities;
- By checking whether all *buffer*'s predictions were equal.

Moreover, in Kinect-NAO configuration, a positive or negative gender-specific sound feedback was implemented on NAO to be given as an output depending on the performance assessment. Once the whole algorithm was established, new acquisitions were performed to test the effectiveness of the new method.

Acquisitions

The new gesture recognition algorithm was tested with new acquisitions. 6 ASD children aged between 4 and 6 were involved in IOGIOCO therapy at CARElab (Computer Assisited Rehabilitation) in Fondazione Don Gnocchi. From the second week of acquisitions, the best model able to classify 19 gestures was tested. In these acquisitions, the therapist supervised the level deciding the gesture to be imitated and taught to the child. NAO performed the gesture selected and pointed at the therapist, thus triggering the evaluation of the therapist's performance by the algorithm. After a temporal window of about 10 seconds, the robot gave a positive or negative sound feedback saying "Well done!" or "Come on, let's do it again!" respectively. The robot pointed at the child and the classifier started its evaluation again.

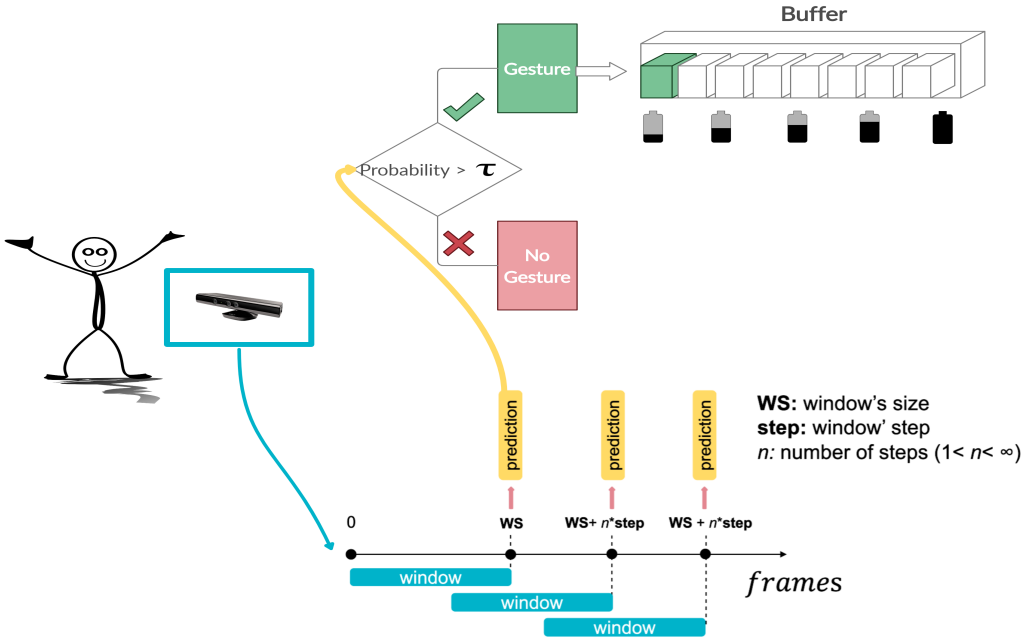

Figure 4: Gesture vs no gesture. Kinect camera captures frames; when the window is filled with the proper number of frames (WS, window size) then a prediction is output. If the highest conditional probability exceeds the threshold, a gesture is detected and the probabilities' prediction vector is saved in the buffer. The window slides of a fixed step and the process re-starts.

Morover, healthy subjects' tests were conducted at Politecnico di Milano. Both Kinect-only and Kinect-NAO configurations were experimented on 2 healthy adults to test gesture recognition performances. 17 selected gestures were correctly performed. In Kinect-NAO acquisitions, actions were performed mimicking the therapy protocol, complying with its timings. So far, *yes* and *no* predictions were discarded even if offline they were properly recognized. *Yes* and *no* are quite challenging movements for the Kinect to capture. In fact, their characterizing movements are described by a small number of joints and a reduced motion range (they involve only the head region) and would need a finer tracking system to be correctly tracked only when intentionally performed.

In order to evaluate the online recognition, Accuracy, F1-score, Precision and Recall were analyzed. Since in clinical acquisitions gestures were performed a different number of times, the class distribution was uneven. For this reason, F1-score was a better measure of the incorrectly classified cases than the accuracy metric.

Results and Discussion

The quantity and the different properties of movements (e.g. duration, range of motion) used in IOGIOCO protocol made the learning process and the online implementation of the recognition algorithm challenging. The overall choices made during the algorithm's development were of great importance to make it robust. The main algorithm settings result are now reported.

Algorithm Specifications

Among the reference points experimented to achieve *translation invariance*, *hip center* turned out to be the most stable joint to this purpose. Since it lies on the human body's sagittal plane, a generalization of the algorithm was possible by the mirroring of movements

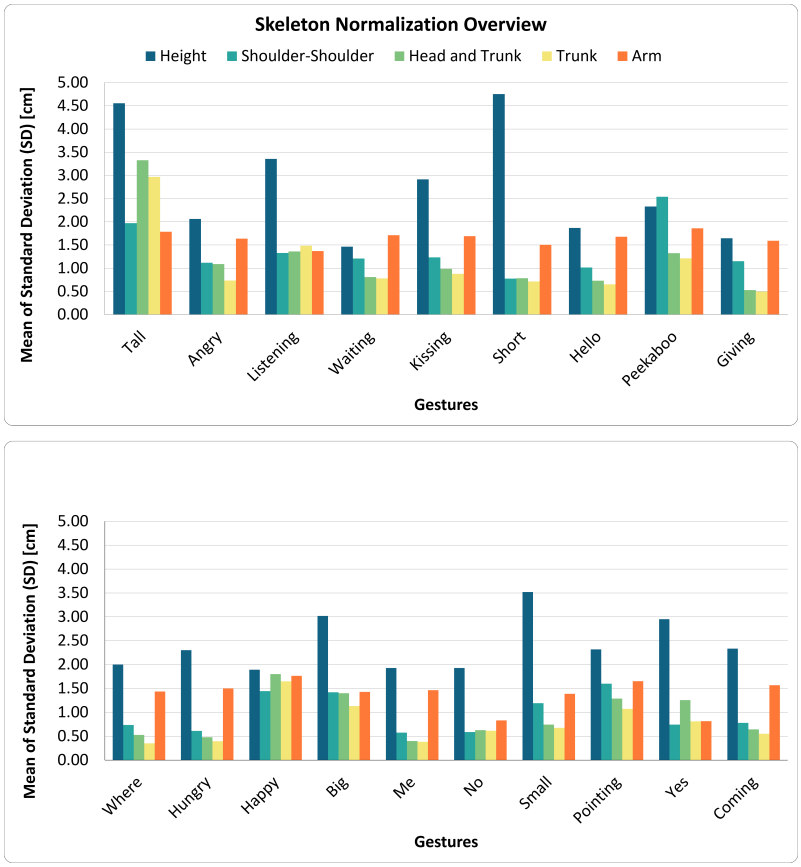

Figure 5: The mean of the Standard Deviation of different normalization's segments for each class in the Expanded Dataset is shown through two column charts.

performed. In this way, the algorithm was able to recognize gestures independently of the dominant hand. For what concerns *user invariance*, results shown in Figure 5 pointed out that the trunk size was the most stable length during the performance of an action for almost all gestures. Thus, it was used as scaling value. From the two charts it's possible to notice that the most unstable segment during the performances of almost all gestures was the height. This was due to Kinect system larger noise behaviour in feet and ankles. Instead, head and trunk and shoulder-shoulder segments had a lower Standard Deviation, since their computation does not involve the bottom part of the body. However, these normalization's segments were not the most stable ones due to head and shoulders movements while performing gestures. The arm length's mean of Standard Deviation was quite the same for all actions, but still high. As a result, the trunk size turned out to be the best solution, since is the least action-involved segment.

After data preparation, the best set of ResNet's hyperparameters was used to train the recognition algorithm. The best model, able to classify all the 19 gestures of IOGIOCO therapy protocol, reached an offline test accuracy of **95%**. Considering the wide gesture set and the different temporal dynamics and duration of actions, this result was encouraging in sight of online recognition.

Online Recognition

In order to analyze Kinect behaviour after NAO connection, FPS was recorded to evaluate

the difference between Kinect-only and Kinect-NAO configurations. In Table 1 FPS's mean and variance are compared. As can be seen, Kinect-NAO configuration slowed down

Table 1: FPS mean and variance for Kinect-only and Kinect-NAO configurations.

Configuration	FPS mean	FPS variance
Kinect-only	50.48	14.65
Kinect-NAO	11	3.56

the frames' capture by the camera. When performing gestures in front of Kinect camera, Kinect-only FPS's mean value was 50.48 fps while, with robot connection, the mean value decreased to 11 fps. Taking into account these results, online settings were properly set reducing the window *size* and *step*.

To monitor the trend of prediction's vector probabilities, *tall*, *hello* and *little* gestures were performed in Kinect-only configuration. Results are shown in Figure 6.

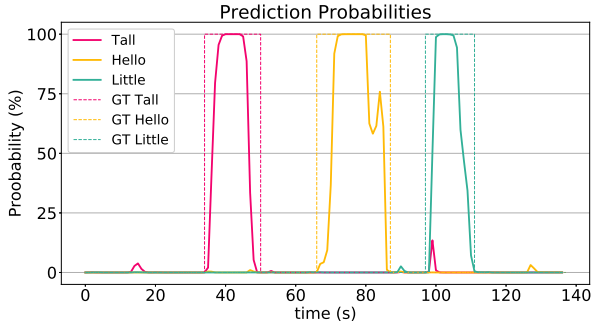

Figure 6: Probability trends of the prediction vector when performing tall, hello and little gestures. The step functions stand for the Ground Truth (GT) i.e. the temporal window in which the gesture was performed.

As expected, when the movements were performed, the gesture-corresponding probability increased. Note that in *little* gesture, since at the beginning of the action both limbs are raised as in *tall* gesture, *tall* probability increased too.

Acquisitions

In Table 2, Accuracy, F1-score, Precision and Recall metrics of all the acquisitions are reported.

Table 2: Metrics scores obtained by the assessment of the algorithm's performances in different sets up.

Configuration	Subjects	Accuracy (%)	F1-score (%)	Precision (%)	Recall (%)
Kinect-only	2 healthy adults	97	97	98	97
Kinect-NAO	2 healthy adults	94	94	95	94
Kinect-NAO	4 ASD children	82	83	89	82

It is worth to notice that recall scores are lower with respect to precision ones. This means that there were reduced chances for an incorrect gesture to be recognized as a correct one, which may be beneficial for therapy sessions.

Kinect-only acquisitions on two healthy subjects resulted in an overall accuracy of **97%** and an F1 score of **97%** for the 17 gestures selected. Instead, with Kinect-NAO

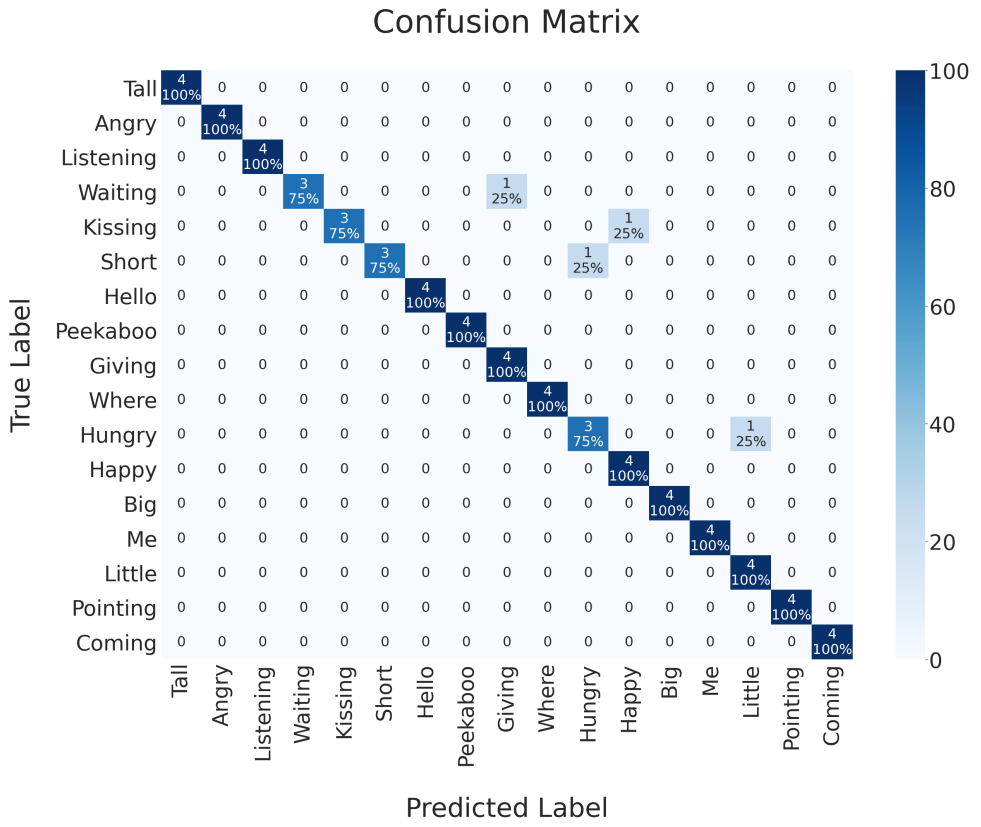

Figure 7: Confusion matrix of healthy adult subjects' acquisitions in Kinect-NAO configuration.

configuration, acquisitions resulted in an overall accuracy of **94%**. Confusion matrix of Kinect-NAO configuration for healthy subject acquisitions is shown Figure 7. As can be seen, *waiting* gesture was confused with *giving*: in fact, the two movements have a similar motion range. Moreover, actions like *short* or *giving* are similar gestures which can be easily mistaken when performed by different subjects. For what concerns *kissing* gesture mistaken with *happy*, keypoints' files were analysed by plotting their joints' coordinates mimicking skeleton movements. It turned out that joints were captured by Kinect in a wrong position, similar to *happy* gesture. The other two mistaken gestures highlighted the importance of timings: the preformed movements started few seconds after NAO pointed and the algorithm analyzed subject's position before the actual execution of the gesture. These results were promising, but it has to be taken into account that the subjects were healthy adults performing gestures in a precise way. Considering the wide gesture set and the different temporal dynamics and duration of actions, outcomes were encouraging in sight of clinical applications.

For what concerns acquisitions in IOGIOCO therapy, during the first 4 weeks 2 out of 6 children successfully familiarized with NAO, accessing Level 2. The other 4 were also able to reach Level 3 to test the recognition algorithm. Taking into account the broad spectrum of conditions of ASD, depending on the child different engagement levels were detected. NAO's feedback was able to engage children's attention, thus increasing their interaction with the therapist and the robot itself. On the other hand, sometimes children lacked of interest in interacting with NAO, thus, in these cases, it was difficult for them

to keep up with the therapy's exercises.

In the 4 children's clinical acquisitions, F1-score resulted **83%**. This F1-score was lower than the **94%** reached with the healthy subjects' acquisitions, but it has to be pointed out that the net was trained on a dataset mainly composed by healthy subjects (only 2 ASD adults out of 22), thus challenging the recognition task for ASD users. Moreover, a lower number of acquisitions were done and not all gestures were tested in the clinical context. Figure 8 reports the assessments of children's performances during the therapy. As can be seen from the confusion matrix, almost all actions were correctly recognized by

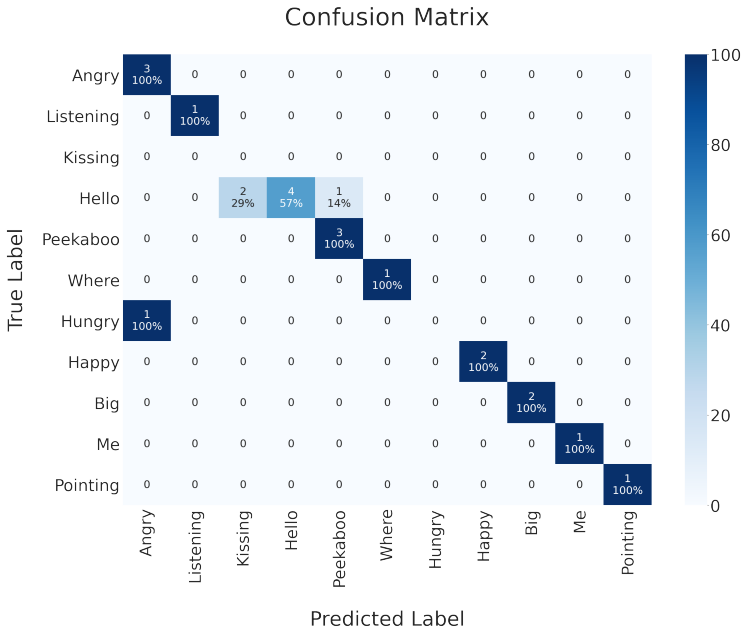

Figure 8: Confusion Matrix of the evaluated children's performances during IOGIOCO therapy in a clinical context.

the algorithm. For what concerns *hungry* gesture, from video analysis resulted that the action was well executed by the child. However, the subsequent raising of the other hand while performing the action made the algorithm recognize a double handed gesture.

Conclusions

This thesis demonstrated successfully the use of a gesture recognition algorithm for the purpose of increasing ASD children engagement and empowering gestures' learning by means of a straightforward and robust feedback system within interactive games. The classification system was tested on 2 healthy subjects and 4 children as part of IOGIOCO therapy. Since ASD has a wide variation in type and severity of symptoms people can experience, children had different ways of approaching the therapy, thus NAO. For this reason, depending on the child, different engagement levels were detected. When children were totally into the therapy, a robot feedback increased their attention and happiness too. Otherwise, with a poor level of engagement, children struggled to comply with IOGIOCO timings. This cases demonstrated the need of improving time settings by which the recognition task starts. Another important aspect is the type of feedback the robot provides. Customized stimuli may be more effective at eliciting skills learning. In point of that, child-specific feedback might empower children's social interaction. In order to

improve the recognition, future work should also aim to update the existing dataset with the collected acquisitions of both healthy subjects and ASD children. In this way, the algorithm would learn to identify gestures differently performed by these users and could be tested on more subjects. Furthermore, the implementation should also consider the next challenging protocol's levels, in which gesture teaching is inserted in a story-telling scenario. Up to date, there is no unequivocal evidence of the effectiveness of this therapy. Therefore, a Randomized Controlled Trial (RCT) would reduce biases when testing this treatment's efficacy.

.

Sommario

Introduzione
Come riportato dall'Organizzazione Mondiale della Sanità, i Disturbi dello Spettro Autistico (DSA) coprono una serie di deficit neuroevolutivi caratterizzati da difficoltà nello stabilire relazioni sociali, nella comunicazione e nel linguaggio, con la presenza di comportamenti ripetitivi e stereotipati. Si stima che nel mondo 1 bambino su 160 abbia DSA [1]. L'intervento nella prima infanzia è fondamentale per promuovere lo sviluppo e gli effetti positivi a lungo termine. I bambini con DSA possono avere difficoltà a comunicare in modo non verbale, ad esempio attraverso gesti, contatto visivo ed espressioni facciali [2].

La terapia standard si basa su interventi in ambito psico-sociale fondati sull'evidenza medica che cercano di adattarsi all'esigenza di ogni bambino. Le terapie più recenti si basano sulla Human-Robot Interaction (HRI) poiché i bambini con DSA focalizzano più facilmente la loro attenzione su sistemi prevedibili come i robot, piuttosto che sugli esseri umani. I bambini con DSA si sentono più a loro agio con i robot rispetto agli esseri umani, che, invece, possono risultare difficili da capire [3]. La modalità "supervised autonomy", in cui il robot umanoide lavora in modo indipendente sotto la guida di un supervisore è nota come RET [4]. In questo modo l'interazione terapista-robot-bambino è favorita, facilitando la comunicazione tra il terapista e il bambino. Come osservato da Sial et al. nel loro lavoro [5], un approccio collaborativo basato su giochi interattivi tra un robot e bambini con DSA ha portato a risultati positivi in termini interazione sociale, comunicazione e attenzione congiunta. Diversi studi hanno dimostrato che i deficit motori sono una comorbidità prevalente nei soggetti con DSA [6], anche se attualmente non sono inclusi nei criteri diagnostici dell'autismo. McAuliffe et al. [7] hanno osservato come queste alterate capacità motorie potrebbero essere collegate a atipiche capacità di apprendimento. In tal senso, i loro risultati supportano l'ipotesi che deficit nell'apprendimento tramite imitazione di gesti possano avere un impatto sullo sviluppo sociale e motorio, poiché l'apprendimento per imitazione è uno dei primi metodi con cui gli esseri umani imparano. Di conseguenza, l'insegnamento tramite imitazione di gesti nelle RET potrebbe migliorare le abilità sociali del bambino così come lo spontaneo utilizzo di gesti [2]. RET carattrizzate da una maggiore autonomia del robot sono importanti per alleggerire il carico di lavoro del terapista e fornire terapie consistenti. In questo contesto, gli algoritmi di riconoscimento di gesti possono essere utilizzati per classifcare l'azione svota dal bambino, in modo tale che il robot ne valuti l'esecuzione tramite feedback. Un feedback appropriato aumenterebbe il coinvolgimento dei bambini, rafforzando così la robustezza della terapia.

Il riconoscimento delle azioni si basa su sistemi di acquisizione in grado di tracciare i movimenti del corpo umano. I bambini con DSA sono particolarmente sensibili al tocco

[8], quindi sistemi indossabili non sono utilizzabili anche se veloci, robusti e ricevono informazioni direttamente dai movimenti degli utenti. Di conseguenza, molti approcci prevedono l'utilizzo di dati RGB o di profondità da video sequenze o una combinazione di questi due (RGB-D) per riconoscere le azioni. Tuttavia, i sistemi di acquisizione video rendono il riconoscimento un compito arduo a causa di molti fattori quali occlusioni del campo visivo della fotocamera, posizionamento della stessa e illuminazione. Tra i sistemi di acquisizione video, le telecamere di profondità a basso costo come Microsoft Kinect v2 sono in grado di fornire un potente metodo di tracciamento del corpo umano in tempo reale [9]. Di conseguenza, i dati relativi allo scheletro umano acquisiti dalla Kinect (coordinate spaziali delle articolazioni) possono essere utilizzati nell'ambito del riconoscimento di gesti.

Gli approcci basati sul deep learning hanno ottenuto risultati promettenti nelle attività di classificazione [10]. Nell'ambito del riconoscimento di azioni, le CNN sono le più utilizzate. In particolare, le ResNet, che si basano sull'apprendimento di funzioni di errore (i.e. residuali), sono una buona soluzione per estrarre con precisione le caratteristiche rilevanti dai dati biomeccanici, consentendo un rapido training della rete e risolvendo il "vanishing gradient problem".

Poiché le reti imparano a riconoscere dai dati con cui vengono allenate, i dataset di training influiscono sul loro comportamento. I più utilizzati sono dataset pubblici e raramente creati da zero. I loro set di gesti sono solitamente selezionati da azioni della vita quotidiana o sono specifici di un particolare contesto (come il Gaming 3D Dataset [11]). Le parti del corpo coinvolte nelle azioni possono comprendere un singolo arto, solo la parte superiore/inferiore del corpo o l'intero scheletro.

L'obiettivo principale di questo progetto di tesi è trovare un metodo adeguato per riconoscere automaticamente i gesti all'interno di una terapia robotica (IOGIOCO) per bambini con DSA. Con l'integrazione dell'algoritmo di riconoscimento, l'efficacia del trattamento aumenta, mentre si alleggerisce il carico di lavoro del terapeuta nell'interazione triadica terapeuta-robot-bambino. L'obiettivo finale è implementare e testare un algoritmo online in un'applicazione clinica per essere in grado di aiutare il terapeuta, promuovendo l'apprendimento dei bambini.

Metodi

La terapia robotica IOGIOCO si basa su giochi di imitazione interattivi tra il robot umanoide NAO e i bambini con DSA. Comprende 5 livelli e una valutazione finale e prevede l'insegnamento di gesti comunicativi selezionati da attività di vita quotidiana. I 19 gesti del protocollo sono: *alto*, *arrabbiato*, *ascolta*, *aspetta*, *bacio*, *basso*, *dare*, *dove*, *fame*, *io*, *cuccù*, *felice*, *sì*, *no*, *grande*, *ciao*, *piccolo*, *puntare*, *vieni* (Figura 9). L'algoritmo di riconoscimento è inserito nella terapia dal Livello 3 in poi.

Il workflow dell'algoritmo di riconoscimento dei gesti ha coinvolto l'acquisizione dei dati, l'elaborazione dei dati e la generazione di pose features per facilitare il processo di apprendimento del classificatore, la classificazione con reti neurali e l'implementazione online (Figura10).

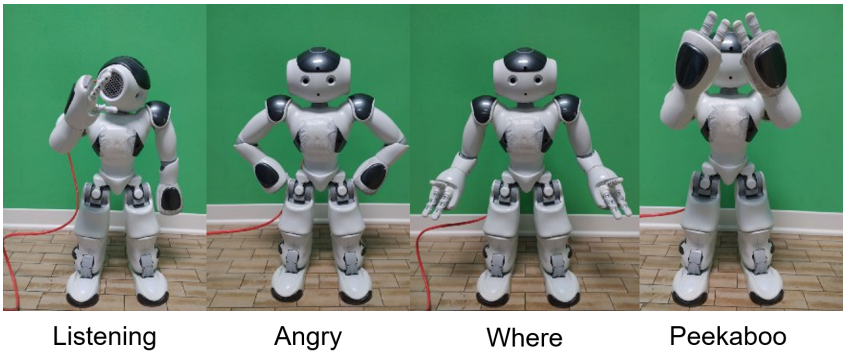

Figura 9: Alcuni gesti eseguiti da NAO: ascolta, arrabbiato, dove cuccù e bacio.

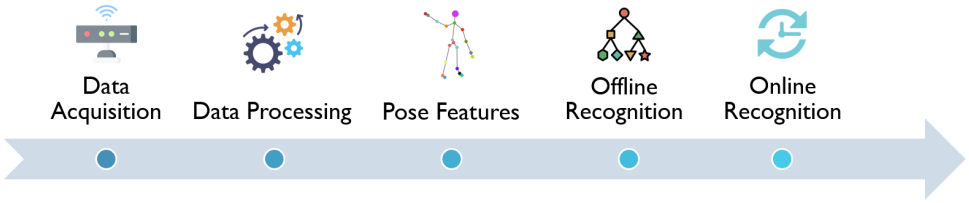

Figura 10: Workflow dell'algoritmo proposto.

Acquisizione di dati

Il metodo proposto utilizza la camera Kinect per acquisire dati poiché è necessario un sistema di acquisizione video non intrusivo per monitorare i movimenti dei bambini e del terapeuta. È in grado di catturare keypoints (coordinate 3D delle articolazioni) consentendo un'analisi spaziale di ogni gesto.

Tre diversi dataset sono stati creati e utilizzati per lo sviluppo dell'algoritmo di riconoscimento dei gesti e sono di seguito elencati. Tra questi, gli ultimi 2 sono stati segmentati manualmente.

1. *Subsampled Healthy Dataset*: dataset con 5 gesti – *piccolo*, *ciao*, *puntare*, *vieni* e *sì*. 18 soggetti adulti sani hanno eseguito queste azioni con una ripetizione.

2. *Healthy Dataset*: dataset di 14 gesti composto da 18 soggetti sani, 9 adulti e 9 bambini. Il set di gesti includeva *alto*, *arrabbiato*, *ascolta*, *aspetta*, *bacio*, *basso*, *ciao*, *dare*, *dove*, *fame*, *grande*, *piccolo*, *puntare* e *vieni*.

3. *Expanded Dataset*: dataset di 19 gesti composto da 22 soggetti, di cui 2 adulti con DSA. Solo 11 soggetti del dataset hanno eseguito ogni tipo di azione. Il set completo di gesti includeva *alto*, *arrabbiato*, *ascolta*, *aspetta*, *bacio*, *basso*, *dare*, *dove*, *ciao*, *cuccù*, *fame*, *felice*, *grande*, *io*, *no*, *piccolo*, *puntare*, *sì*, *vieni*.

Elaborazione dei dati

Le informazioni estratte dalla Kinect sono state elaborate. Un filtraggio è stato necessario per ridurre il rumore di cui la Kinect è affetta. Poiché le coordinate erano riferite alla telecamera, il sistema di riferimento è stato centrato rispetto al soggetto stesso. Sono stati testati diversi punti di riferimento per ottenere *invarianza alla traslazione*. Inoltre, poiché in questo progetto sono stati coinvolti adulti e bambini sani e soggeti con DSA con

strutture fisiche differenti, è stata utilizzata una normalizzazione per ottenere *invarianza tra soggetti*. È stata eseguita una normalizzazione fotogramma per fotogramma sui dati a causa della bassa risoluzione spaziale sui dati di profondità [12] della Kinect e della posizione indefinita del soggetto davanti alla telecamera, che potrebbero influenzare il valore di normalizzazione non riflettendo le dimensioni reali del corpo umano. Questo approccio ha permesso un aumento della robustezza del metodo di normalizzazione. Tuttavia, una normalizzazione fotogramma per fotogramma potrebbe cambiare la lunghezza del segmento di normalizzazione, calcolato come distanza euclidea, da un fotogramma a quello adiacente: questo è vero soprattutto quando si tratta, ad esempio, del gesto *sì* o di qualsiasi gesto che coinvolga un piegamento in avanti del busto. Per indagare a fondo quanto questi segmenti cambiassero da un fotogramma all'altro, per ogni classe di gesti è stata calcolata la lunghezza media del segmento sulla sequenza video in ogni campione. Quindi un confronto tra la variazione dei segmenti durante l'azione per ogni classe di gesti è stato ottenuto mediante deviazione standard.

Inoltre, i dati Kinect sono stati organizzati secondo la biomeccanica al fine di evidenziare la cinematica che caratterizza un particolare gesto in un unico campione. Poiché la parte inferiore del corpo non era essenziale nel set di gesti, sono stati conservati solo i segmenti della parte superiore del corpo. Per ottenere una rappresentazione efficace del gesto eseguito e per mantenere le caratteristiche del movimento locale, le coordinate delle articolazioni delle sequenze scheletriche della parte superiore del corpo sono state suddivise in set corporei (due braccia e un tronco). I set sono stati riorganizzati dall'alto verso il basso secondo la struttura fisica del corpo umano (prima testa e tronco, di seguito braccio destro e sinistro). Pertanto, sono state generate matrici 3D per descrivere le azioni concatenando insieme ogni fotogramma del movimento, composto da array di coordinate 3D delle articolazioni.

Pose Features

Dopo l'elaborazione dei dati, è stato effettuato un focus sul significato degli input della rete neurale generando *pose features* in grado di descrivere la cinematica delle azioni eseguite. La corretta preparazione di queste features potrebbe in qualche modo guidare il processo di apprendimento del classificatore [13]. Per ottenere una *pose feature*, tutte le coordinate 3D (x_k, y_k, z_k) di ogni frame F_t in una sequenza sono state scalate tramite una funzione di normalizzazione $\mathbf{N}(\cdot)$:

$$\begin{aligned}(x'_k, y'_k, z'_k) &= \mathbf{N}(x_k, y_k, z_k) \\ x'_k &= \frac{(x_k - x_{min})}{(x_{max} - x_{min})}, \\ y'_k &= \frac{(y_k - y_{min})}{(y_{max} - y_{min})}, \\ z'_k &= \frac{(z_k - z_{min})}{(z_{max} - z_{min})},\end{aligned} \tag{3}$$

dove (x'_k, y'_k, z'_k) sono le coordinate normalizzate del k-esimo keypoint, $\mathbf{c_{max}}(x_{max}, y_{max}, z_{max})$ e $\mathbf{c_{min}}(x_{min}, y_{min}, z_{min})$ sono le coordinate di scala. Per standardizzare ogni classe di azione, sono state sperimentate diverse tecniche di *normalizzazione dei gesti*. I keypoints sono stati ridimensionati sfruttando una normalizzazione *gesto-indipendente*. Pertanto, le coordinate massime e minime di ciascun canale (x, y, z) di ogni sequenza di movimenti sono state rilevate qualunque gesto eseguito e utilizzate come valori di scala. Inoltre, $\mathbf{c_{max}}$

e $\mathbf{c_{max}}$ sono state selezionate indipendentemente dalle articolazioni del corpo, quindi in un volume di controllo comprendente il corpo intero. Queste coordinate di scala avrebero potuto essere calcolate rispetto all'intero dataset utilizzato, ottenendo così valori di normalizzazione diversi per ciascuna classe nel set di gesti. Tuttavia, questo approccio sarebbe stato specifico per il dataset, mentre una normalizzazione *gesto-indipendente* ha reso il sistema *dataset-indipendente*.

Quando si usano le ANN e le uniche informazioni sul movimento disponibili sono i dati scheletrici, una rappresentazione intermedia delle sequenze scheletriche può aiutare nell'elaborazione dei dati e nella comprensione dei campioni da cui la rete deve imparare. Pertanto, *RGB pose features* sono state ottenute trasformando lo spazio delle coordinate nello spazio colore RGB, scalando ciascuna coordinata nell'intervallo [0, 255]. In questo modo, la cinematica di ogni azione è stata preservata ed evidenziata da una nuova rappresentazione visiva. Durante un movimento, uno spostamento nella direzione x è rappresentato da una variazione della quantità di rosso, mentre uno spostamento nella direzione y o z corrisponde a un cambiamento nel verde o blu rispettivamente. La figura 11 mostra il modo in cui viene generata una RGB pose feature.

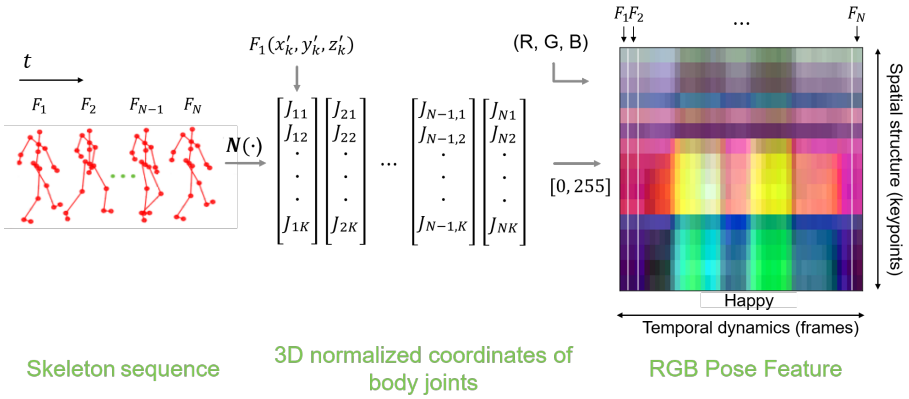

Figura 11: Dalle sequenze scheletriche a RGB pose feature: ogni fotogramma F_t di una sequenza è stato trasformato in un array 3D per essere concatenato in una matrice insieme agli altri. N indica il numero di fotogrammi in ciascuna sequenza e K indica il numero di keypoints in ogni fotogramma. Quindi, dopo la normalizzazione delle coordinate, viene applicato un ridimensionamento nello spazio colore RGB, ottenendo così una singola RGB pose feature che rappresenta il movimento. Sulla dimensione orizzontale è mostrata la dinamica temporale, mentre la struttura spaziale (keypoints) è rappresentata su quella verticale.

A partire da questa rappresentazione scheletrica, i dati sono stati ulteriormente elaborati:

- Interpolazione temporale e ridimensionamento: poiché le CNN richiedono che gli input abbiano la stessa lunghezza, sono stati testati diversi modi di interpolazione e ridimensionamento;

- Enhanced Action Images: una tecnica di miglioramento del contrasto locale è stata utilizzata per evidenziare ulteriormente le caratteristiche del movimento;

- Data Mirroring: poiché i gesti utilizzati nel training del modello sono stati eseguiti con il braccio destro, il data mirroring è stato in grado di rendere il classificatore indipendente dalla mano utilizzata per eseguire l'azione e di aumentare il dataset.

Classificazione
È stata implementata una rete Residual Network (ResNet) poiché è in grado di costruire una rete neurale profonda senza il rischio di comprometterne le prestazioni. Ciò è possibile grazie alle "skip connections", che permettono alla rete di saltare il training di uno o più layer. Inoltre, una funzione Softmax viene spesso utilizzata nell'ultimo layer della rete. La funzione Softmax trasforma l'output numerico dell'ultimo layer lineare di una rete neurale di classificazione multiclasse in un vettore di N probabilità, dove N è il numero di classi. Perciò, l'output della ResNet è caratterizzato da un vettore che rappresenta la distribuzione di probabilità di tutti i 19 gesti. Le pose features sono state utilizzate come input. Al fine di valutare l'algoritmo di riconoscimento, è stato sfruttato un metodo di convalida incrociata dei soggetti Leave-P-Out per il dataset più completo, l' Expanded Dataset, con 19 gesti. In questo modo, P su N soggetti nel dataset sono stati utilizzati per il test e P−N per il training e la convalida del modello (P = 2 soggetti e N = 11 soggetti dell'Expanded Dataset). Sono stati testati diversi iperparametri della rete per ottenere i migliori risultati di riconoscimento possibili.
Riconoscimento online
Una volta ottenuto l'algoritmo offline, esso è stato implementato online. L'implementazione online comprendeva due passaggi:

- Configurazione solo-Kinect: il modello è stato impostato e testato sul flusso di dati continuo catturato dalla telecamera;

- Configurazione Kinect-NAO: il modello è stato impostato e testato con il robot.

Sono state sperimentate diverse impostazioni per tenere conto del comportamento della Kinect con la connessione del robot. Per questo motivo, gli FPS della telecamera Kinect sono stati valutati in entrambe le configurazioni per monitorare la frequenza dei fotogrammi, poiché l'attività di riconoscimento si basava sull'acquisizione dei dati delle pose dello scheletro.

Per sfruttare l'algoritmo di riconoscimento in una classificazione in tempo reale è stata utilizzata una finestra scorrevole. Le pose features sono state calcolate e analizzate dal classificatore su una determinata finestra, caratterizzata da due parametri di configurazione: *dimensione* e *step*. Una finestra con *dimensione* fissa in termini di numero di fotogrammi è stata utilizzata sul flusso di dati continuo catturato dalla telecamera. Sono state sperimentate diverse configurazioni per evitare il più possibile il ritardo tra la performance dell'azione e l'output della classificazione. Per rilevare la presenza di un gesto, l'output di probabilità condizionale più alto dal Softmax layer del classificatore è stato confrontato con una soglia $\tau \ \ in$ [0,1]:

$$stato = \begin{cases} gesto, & \text{se } probabilità > \tau \\ nessun - gesto, & \text{se } probabilità < \tau \end{cases} \tag{4}$$

Quando la soglia di rilevamento è stata superata, il vettore di previsione delle probabilità è stato salvato in un *buffer*. Quindi, la finestra si è mossa di uno *step* prefissato prima di

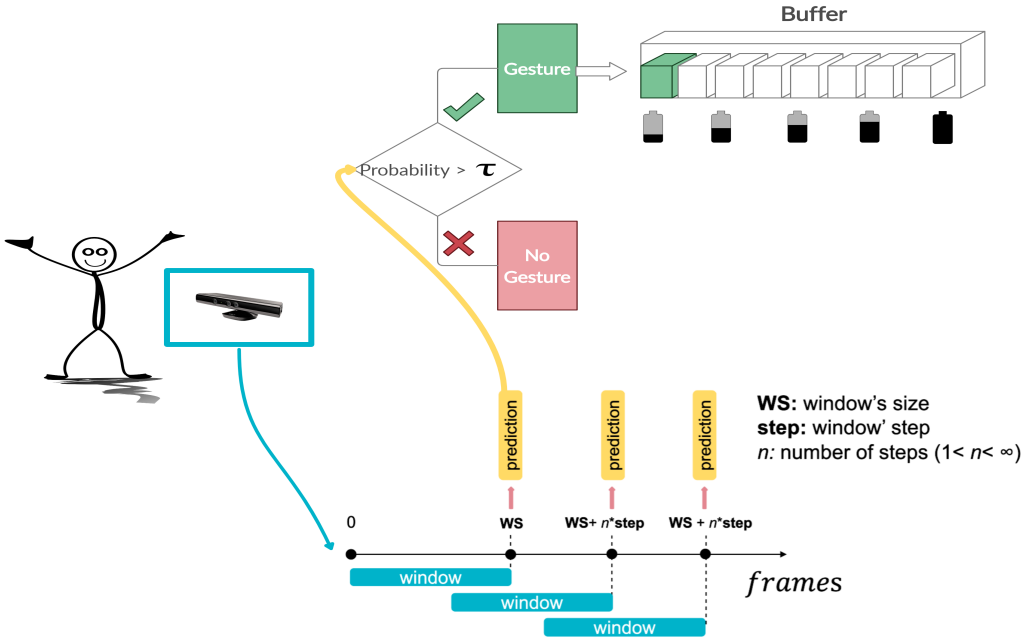

Figura 12: Gesto vs nessun gesto. La fotocamera Kinect acquisisce i fotogrammi; quando la finestra è piena con il numero corretto di frame (WS, dimensione della finestra), viene emessa una previsione. Se la probabilità condizionale più alta supera la soglia, viene rilevato un gesto e il vettore di previsione delle proabilità viene salvato nel buffer. La finestra scorre di uno step specifico e il processo si ripete.

prevedere di nuovo (Figura 12). Una volta che il *buffer* è stato riempito con N vettori di previsione, l'algoritmo ha identificato il gesto eseguito con uno dei due possibili metodi:

- Calcolando la media delle probabilità dei vettori di previsione del *buffer*;
- Controllando se tutte le previsioni del *buffer* fossero uguali.

Inoltre, nella configurazione Kinect-NAO, è stato implementato sul robot un feedback sonoro positivo o negativo da fornire come output a seconda della valutazione delle prestazioni. Una volta implementato l'intero algoritmo, sono state eseguite nuove acquisizioni per testare l'efficacia del nuovo metodo.

Acquisizioni

Infine, il nuovo algoritmo di riconoscimento di gesti è stato testato con nuove acquisizioni. 6 bambini con DSA di età compresa tra i 4 ei 6 anni sono stati coinvolti nel protocollo terapeutico IOGIOCO al CARElab (Computer Assisited Rehabilitation) della Fondazione Don Gnocchi. Dalla seconda settimana di acquisizioni è stato sperimentato il miglior modello in grado di riconoscere 19 gesti. In queste acquisizioni, il terapista supervisionava il livello decidendo il gesto da imitare e da insegnare al bambino. NAO eseguiva il gesto selezionato e puntava verso il terapista, innescando così la valutazione della performance del terapeuta da parte dell'algoritmo. Dopo una finestra temporale di circa 10 secondi, il robot forniva un feedback sonoro positivo o negativo dicendo "Ben fatto!" o "Dai, facciamolo di nuovo!" rispettivamente. NAO indicava il bambino e il classificatore iniziava di nuovo la sua valutazione.

Inoltre, al Politecnico di Milano sono stati condotti test su soggetti sani. Entrambe le configurazioni solo-Kinect e Kinect-NAO sono state sperimentate su 2 adulti sani per

testare le prestazioni di riconoscimento dei gesti. Nelle acquisizioni Kinect-NAO sono state eseguite le azioni imitando il protocollo della terapia rispettandone i tempi. 17 gesti selezionati sono stati correttamente eseguiti. Finora, le previsioni di *sì* e *no* sono state scartate anche se offline erano state correttamente riconosciute. *Sì* e *no* sono movimenti piuttosto impegnativi da tracciare per la Kinect . Infatti, i loro movimenti caratterizzanti sono descritti da un numero ridotto di articolazioni e da un raggio di movimento ridotto (coinvolgono solo la regione della testa) e richiederebbero un sistema di tracciamento più fine per essere correttamente tracciati solo se intenzionalmente eseguiti.

Al fine di valutare il riconoscimento online, sono stati analizzati Accuratezza, F1-score, Precisione e Recall. Poiché nelle acquisizioni cliniche i gesti venivano eseguiti ognuno un numero diverso di volte, la distribuzione delle classi era irregolare. Per questo motivo, l'F1-score è stato una misura più rappresentativa dei casi classificati in modo errato rispetto all'accuratezza.

Risultati e Discussione

La quantità e le diverse proprietà dei movimenti (ad esempio durata, range di movimento) utilizzate nel protocollo IOGIOCO hanno reso difficile il processo di apprendimento e l'implementazione online dell'algoritmo di riconoscimento. Le scelte complessive fatte durante lo sviluppo dell'algoritmo sono state di grande importanza per renderlo robusto. Vengono ora riportati i risultati delle impostazioni principali dell'algoritmo.

Specifiche dell'algoritmo

Tra i punti di riferimento sperimentati per ottenere l' *invarianza di traslazione*, il *centro dell'anca* si è rivelato l'articolazione più stabile a questo scopo. Poiché si trova sul piano sagittale del corpo umano, il mirroring dei movimenti eseguiti ha permesso una generalizzazione dell'algoritmo. In questo modo, l'algoritmo è stato in grado di riconoscere i gesti indipendentemente dalla mano dominante. Per quanto riguarda l'*invarianza tra soggetti*, i risultati mostrati in Figura 13, hanno evidenziato che la dimensione del tronco era la lunghezza più stabile durante l'esecuzione di un'azione per quasi tutti i gesti. Pertanto, è stato utilizzato come valore di scala. Dai due grafici è possibile notare che il segmento più instabile durante l'esecuzione di quasi tutti i gesti è stato quello dell'altezza. Ciò era dovuto al maggiore rumore del sistema Kinect per quanto riguarda le articolazionni dei piedi e delle caviglie. I segmenti della testa-tronco e spalla-spalla avevano invece una deviazione standard inferiore, poiché il loro calcolo non coinvolge la parte inferiore del corpo. Tuttavia, questi segmenti di normalizzazione non sono risultati i più stabili a causa dei movimenti della testa e delle spalle durante l'esecuzione dei gesti. La deviazione standard media della lunghezza del braccio era più o meno la stessa per tutte le azioni, ma comunque alta. Di conseguenza, la dimensione del tronco si è rivelata la soluzione migliore, poiché è il segmento meno coinvolto nell'azione.

Dopo la preparazione dei dati, il miglior set di iperparametri per la ResNet è stato utilizzato per allenare l'algoritmo di riconoscimento. Il miglior modello, in grado di classificare tutti i 19 gesti del protocollo terapeutico IOGIOCO, ha raggiunto una precisione di test offline di **95%**. Considerando l'ampio set di gesti e le diverse dinamiche temporali e durate delle azioni, questo risultato è stato incoraggiante in vista del riconoscimento online.

Riconoscimento Online

Per analizzare il comportamento della Kinect dopo la connessione del robot, sono stai analizzati i FPS per valutare la differenza tra le configurazioni solo-Kinect e Kinect-NAO.

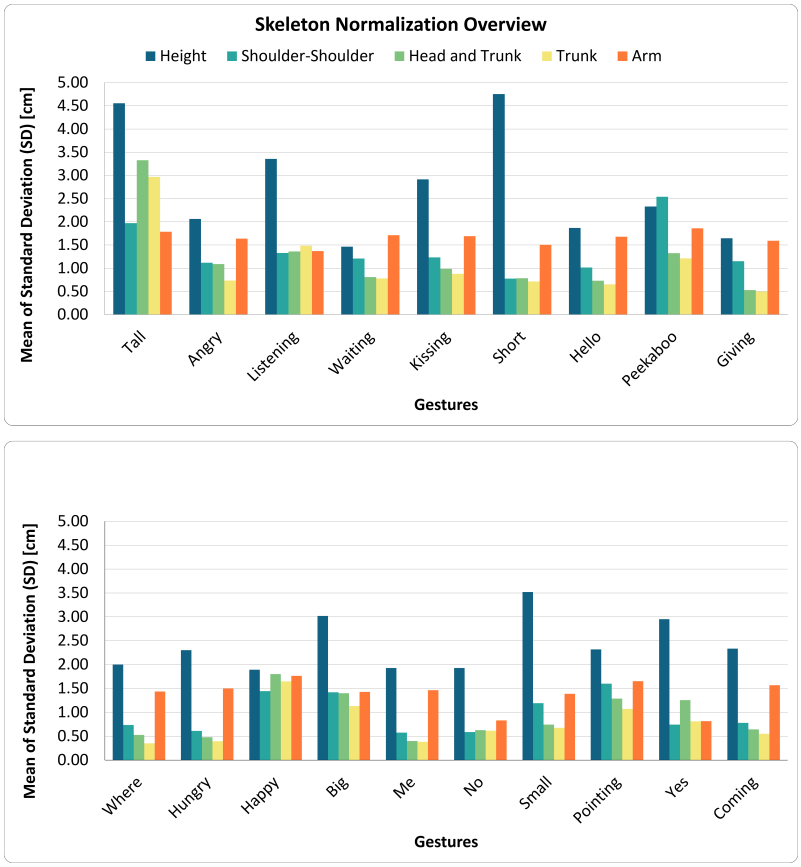

Figura 13: La media della deviazione standard lungo i frame di diversi segmenti di normalizzazione su tutti i campioni nell'Expanded Dataset è mostrata attraverso due grafici a colonne.

Nella tabella 3 vengono confrontate la media e la varianza dei FPS. Come si può vedere,

Tabella 3: Media e varianza FPS per le configurazioni solo Kinect e Kinect-NAO.

Configurazione	**FPS mean**	**FPS variance**
solo Kinect	50.48	14.65
Kinect-NAO	11	3.56

la configurazione Kinect-NAO ha rallentato l'acquisizione dei fotogrammi da parte della telecamera. Durante l'esecuzione dei gesti, il valore medio di FPS di Kinect era di 50.48 fps mentre, con la connessione del robot, il valore medio scendeva a 11 fps. Tenendo conto di questi risultati, i settung online sono stati impostati adeguatamente riducendo *dimensione* e *step* della finestra.

Per monitorare l'andamento delle probabilità dei vettori di previsione, i gesti *alto*, *ciao* e *piccolo* sono stati eseguiti nella configurazione solo-Kinect. I risultati sono mostrati nella Figura 14.

Come previsto, quando sono stati eseguiti i movimenti, la probabilità corrispondente al gesto è aumentata. Nota che nel gesto *piccolo*, poiché all'inizio dell'azione entrambi gli arti sono sollevati come nel gesto *alto*, aumenta anche la probabilità di *alto*. *Acquisizioni*

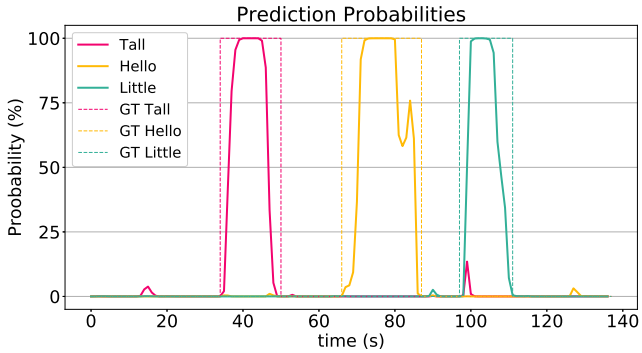

Figura 14: Andamento delle probabilità dei vettori di previsione quando si eseguono i gesti alto, ciao e piccolo nella configurazione solo-Kinect.

Nella tabella 4, vengono riportate le metriche Accuratezza, F1-score, Precisione, Recall di tutte le acquisizioni.

Tabella 4: Punteggi delle metriche ottenuti dalla valutazione delle prestazioni dell'algoritmo in diverse configurazioni.

Configurazione	**Soggetti**	**Accuratezza (%)**	**F1-score (%)**	**Precisione (%)**	**Recall (%)**
solo-Kinect	2 adulti sani	97	97	98	97
Kinect-NAO	2 adulti sani	94	94	95	94
Kinect-NAO	4 bambini DSA	82	83	89	82

Si può notare che i punteggi di Recall sono inferiori rispetto a quelli di Precisione. Ciò significa che c'erano ridotte possibilità di riconoscere un gesto errato come corretto, il che può essere utile per le sessioni di terapia.

Dalle acquisizioni solo-Kinect su due soggetti sani si è ottenuta un' acuratezza complessiva di **97%** e un F1-score di **97%** per i 17 gesti selezionati. Invece, con la configurazione Kinect-NAO, l'accuratezza complessiva era di **94%**. La matrice di confusione della configurazione Kinect-NAO per acquisizioni di soggetti sani è mostrata nella Figura 15.

Come si può vedere, il gesto di *aspetta* è stato confuso con *dare*: i due movimenti, infatti, hanno un range di volume d'azione simile. Inoltre, azioni come *basso* o *dare* sono gesti simili che possono essere facilmente confusi se eseguiti da soggetti diversi. Per quanto riguarda il gesto *bacio* scambiato con *felice*, sono stati analizzati i file dei keypoints tracciando le coordinate delle loro articolazioni che imitano i movimenti dello scheletro. Da questa analisi è risultato che le articolazioni sono state tracciate dalla Kinect in modo errato, simile al gesto *felice*. Gli altri due gesti confusi hanno evidenziato quanto siano importanti le tempistiche: i movimenti eseguiti sono iniziati pochi secondi dopo che il robot ha puntato e l'algoritmo ha analizzato la posizione del soggetto prima dell'esecuzione del gesto vero e proprio. Nell'insieme, i risultati sono promettenti, ma bisogna tener conto che i soggetti erano adulti sani che eseguivano gesti in modo preciso. Considerando l'ampia gamma di gesti e le diverse dinamiche temporali e la durata delle azioni, i risultati sono stati incoraggianti in vista delle applicazioni cliniche.

Per quanto riguarda le acquisizioni nella terapia IOGIOCO, durante le prime quattro settimane 2 bambini su 6 hanno familiarizzato con successo con NAO, accedendo al Livello

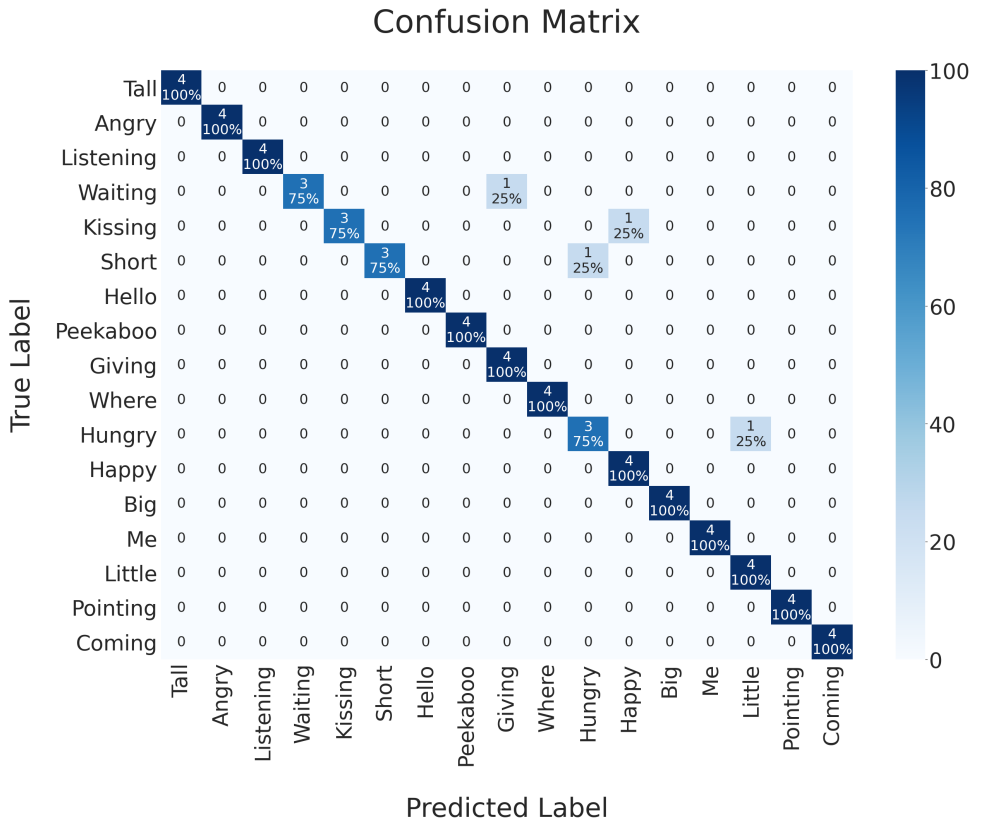

Figura 15: Matrice di confusione delle acquisizioni di soggetti adulti sani nella configurazione Kinect-NAO.

2. Gli altri 4 sono stati anche in grado di raggiungere il Livello 3 per testare l'algoritmo di riconoscimento. Tenendo conto dell'ampio spettro dell'autismo, a seconda del bambino sono stati rilevati diversi livelli di coinvolgimento. Il feedback del robot è stato in grado di promuovere l'attenzione dei bambini, aumentando così la loro interazione con il terapeuta e il robot stesso. D'altra parte, a volte i bambini non erano interessati a interagire con il robot, quindi, in questi casi, è stato difficile per loro tenere il passo con gli esercizi della terapia.

Nelle acquisizioni cliniche di 4 bambini, l' F1-score ha raggiunto **83%**. Questo F1-score era inferiore al **94%** raggiunto con le acquisizioni su soggetti sani, ma va sottolineato che la rete è stata allenata su un dataset composto principalmente da soggetti sani (solo 2 ASD adulti su 22 soggetti), rendendo più complesso il compito di riconoscimento per gli utenti con DSA. Inoltre, è stato effettuato un numero inferiore di acquisizioni e non tutti i gesti sono stati testati nel contesto clinico. La figura 16 riporta le valutazioni delle prestazioni dei bambini durante la terapia.

Come si può vedere dalla matrice di confusione, quasi tutte le azioni sono state correttamente riconosciute dall'algoritmo. Per quanto riguarda il gesto *fame*, dalla video analisi è emerso che l'azione è stata effettivamente ben eseguita dal bambino. Tuttavia, il successivo sollevamento dell'altra mano durante l'esecuzione dell'azione ha fatto sì che l'algoritmo riconoscesse un gesto a due mani.

Conclusioni

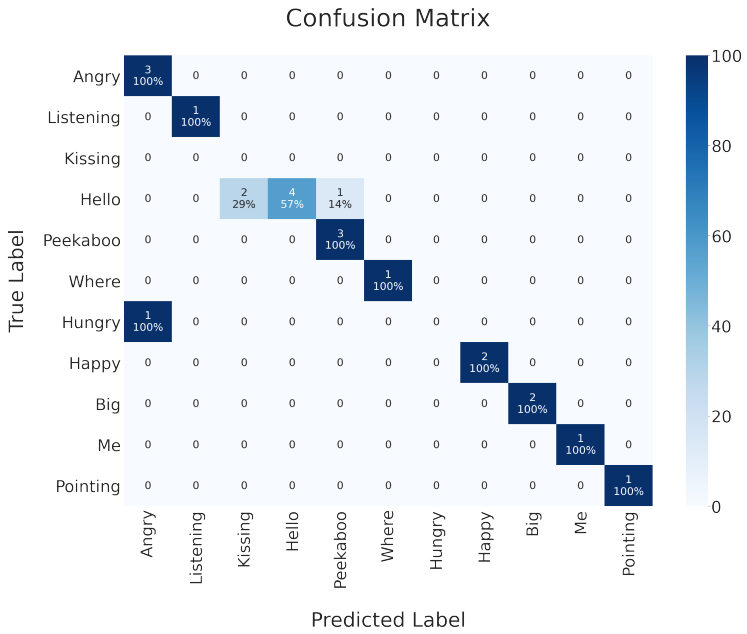

Figura 16: Matrice di confusione delle prestazioni dei bambini valutati durante la terapia IOGIOCO in un contesto clinico.

Questa tesi ha dimostrato con successo l'uso di un algoritmo di riconoscimento dei gesti allo scopo di aumentare il coinvolgimento dei bambini con DSA e promuovere l'apprendimento dei gesti per mezzo di un robusto sistema di feedback all'interno di giochi interattivi. Il sistema di classificazione è stato testato su 2 soggetti sani e 4 bambini come parte del protocollo terapeutico IOGIOCO. Poiché l'autismo è caratterizzato da un ampio spettro nel tipo e nella gravità dei sintomi, i bambini avevano modi diversi di approcciarsi alla terapia IOGIOCO, quindi a NAO. Per questo motivo, a seconda del bambino, sono stati rilevati diversi livelli di coinvolgimento. Quando i bambini erano totalmente coinvolti nella terapia, il feedback di NAO aumentava anche la loro attenzione e felicità. Altrimenti, quando il livello di coinvolgimento nella terapia era scarso, i bambini hanno faticato a rispettare le tempistiche del protocollo terapeutico. Questi casi hanno dimostrato la necessità di migliorare le impostazioni dell'istante in cui inizia l'attività di riconoscimento dell'algoritmo all'interno del protocollo. Un altro aspetto importante è il tipo di feedback fornito dal robot. Gli stimoli personalizzati possono essere più efficaci per promuovere l'apprendimento. In tal senso, feedback specifici per ogni bambino potrebbero potenziarne l' interazione sociale. Per migliorare il riconoscimento, il lavoro futuro dovrebbe anche mirare ad aggiornare il dataset esistente con le acquisizioni raccolte sia di soggetti sani sia di bambini con DSA. In questo modo, l'algoritmo imparerebbe a identificare i gesti diversamente eseguiti da questi utenti e potrebbe essere testato su più soggetti. Inoltre, l'implementazione dovrebbe anche considerare i livelli successivi del protocollo, più complessi e impegnativi, in cui l'insegnamento dei gesti è inserito in uno scenario di narrazione. Ad oggi, non ci sono prove inequivocabili dell'efficacia di questa terapia. Pertanto, uno studio randomizzato controllato (RCT) ridurrebbe gli errori durante il test dell'efficacia di questo trattamento.

Contents

Acronyms

ADHD Attention Deficit Hyperactivity Disorder.

ADL Activities of Daily Living.

ANN Artificial Neural Network.

ASD Autism Spectrum Disorder.

CLAHE Contrast Limited Adaptive Histogram Equalization.

CNN Convolutional Neural Network.

FPS Frames Per Second.

HMM Hidden Markov Model.

KNN K-Nearest Neighbors.

LSTM Long Short Term Memory.

RAT Robot-Assisted Therapy.

RC Robot Coach.

ResNet Residual Network.

RET Robot-Enhanced Therapy.

RNN Recurrent Neural Network.

SVM Support Vector Machine.

TC Therapist Coach.

WoZ Wizard of Oz.

Chapter 1

Introduction

1.1 Motivation

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition characterised by some degree of impaired social behaviour, communication and language and a narrow range of interests and activities that are both unique to the individual and carried out repetitively, as the World Health Organization (WHO) states.

In the past few decades, studies have demonstrated that ASD occurs globally, and that the numbers of recorded cases are rising. There are many possible explanations for this apparent increase, including improved awareness, expansion of diagnostic criteria, better diagnostic tools and improved reporting. It is estimated that worldwide 1 in 160 children has ASD [1].

The aetiology of ASD is still being studied and, despite years of research, a complete understanding of the causative factors is still elusive. Despite the awareness that genetics may play a big role in ASD, a rapidly increasing prevalence suggests a bigger role of environmental factors [14]. Intervention during early childhood is important to promote the optimal development and well-being of people with an ASD.

Gold-standard treatments focus on the delivery of evidence-based psycho-social interventions, such as behavioural treatments and skills training programs. Newest interventions exploit Robot-Assisted Therapy (RAT); in fact, some studies in the literature show that people with ASD are more comfortable with rule-based and predictable systems rather than human beings, which are perceived as hard to understand and even frightening [3]. RAT enables embodied interactions through an increasing in the engagement and attention, a decreasing in social anxiety and the maintenance of simplicity and predictability [4]. Besides possibly experiencing difficulties developing and understanding language skills, children with ASD may have difficulty communicating non-verbally, such as through hand gestures, eye contact, and facial expressions [2]. As observed by Sial et al. in their work [5], a collaborative approach based on interactive games between a robot and ASD children has produced positive results in terms of therapeutic outcomes such as social interaction, communication, joint attention and turn taking. Studies have shown that motor impairments are a prominent comorbidity within the ASD phenotype [6], even though they are not currently included in the diagnostic criteria of autism. McAuliffe et al. [7] explored

how these altered motor ability might be linked to an atypical skills learning. In point of that, their results supports the hypothesis that a poor imitative gestural learning can impact social and motor development, since learning via imitation is a prime method by which humans acquire skills. Also [15] reports that children with ASD exhibit significant impairments both in imitation of gestures as well as in their spontaneous use, which have been found to be related to the development of social interaction. These findings suggest that teaching gesture imitation may improve the child's social skills and even language development [16] as well as spontaneous gesture use [17].

1.2 Goal

This thesis' project focuses on the implementation and testing of an online gesture recognition algorithm to be used in a robot-therapy environment for children with autism. This thesis is integrated in a bigger project promoted by CARELab in Fondazione Don Gnocchi Milano, called IOGIOCO. IOGIOCO robot therapy aims to empower ASD children with gestures meaningful from the communicative point of view (transitive and intransitive gestures). The humanoid robot NAO should react properly to children's movements when interacting with them and support therapist's work in the Therapist-Robot-Child triad. Thus, the principal aim of this work is to develop an automatic method to classify the action performed and assess users' performances. The proper feedback should be a trigger to increase children's engagement while complying with protocol's timings. The final goal is to integrate the algorithm in the therapy to be able to help therapist's work, promoting children's learning.

1.3 Overview

To achieve the goal, a Residual Network (ResNet) was exploited since artificial neural networks are widely used in recognition tasks [10]. Both Offline and Online recognition systems were implemented.

Firstly, the net was designed and tested offline with previous data acquisitions. The chosen method exploits Kinect camera acquisitions since a non-intrusive vision-based capturing system is required to monitor children with autism, which are particularly sensible to the touch, thus wearable systems would be unfeasible. These acquisitions included several gestures performed mainly by healthy subjects. The net had to learn from data first and apply this knowledge to new one then. Inputs' characteristics and processing are essential when dealing with ANN, since they are the most reality-related part. In fact, even though the way nets learn it is not well understood, a proper data preparation will drive the learning process [13]. Thus, the information extracted from Kinect was processed. Filtering was necessary to reduce noise by which Kinect is affected. Since coordinates are referred to the Kinect, the reference system was centered with respect to the subject himself. To achieve user-invariance properties, different frame by frame normalizations were experimented and evaluated by the analysis of the body scaling segments' variation during the performance of the action. After, *pose features* were generated: joints' coordinates of skeleton sequences were rearranged according to the physical structure of the human body to get an effective representation of the action. In addition, when using ANN and the

only available motion features are skeletal data, an intermediate image representation of skeletal sequences can help in data processing and in exploring the samples the net has to learn from. Therefore, *RGB pose features* were obtained by transforming the coordinate space into RGB color space. Starting from this skeleton-based representations, further data processing was implemented.

When data preparation was concluded and properly adapted to this project's goal, the choice of a suitable classification method was essential. Residual learning (based on the learning of error functions) turned out to be a good solution to extract with precision relevant features from biomechanical sequences, because of its fast training process and its ability of resolving the vanishing gradient problem.

Once the algorithm was established offline, an online implementation was designed to be integrated in the robot therapy. Different settings were experimented to comply with protocol timings and to take into account Kinect behaviour with robot connection.

Finally the new gesture recognition algorithm was tested: healthy subjects' tests were conducted at Politecnico di Milano and 6 ASD children were involved in IOGIOCO therapy.

The thesis demonstrated successfully the use of the gesture recognition algorithm for the purpose of increasing ASD children engagement and empowering gestures learning. The results obtained with healthy adults and autistic children were quite encouraging, and pave the way to new developments in the near future.

1.4 Thesis structure

The current thesis is organized as follows: first, the literature related to this work is reviewed and analyzed in Chapter 2. The keywords used in the literature search involved the concepts of autism, robot therapy and gesture recognition.

In Chapter 3, the gesture recognition algorithm is presented. IOGIOCO protocol and the datasets used are described. The methods used in this work are outlined, starting with Data processing and Pose Feature overview. Classification, Online Recognition and Acquisitions are then detailed.

In Chapter 4 Data processing's results and offline models are presented. Furthermore, Online settings and two healthy subjects' acquisitions results are reported. Moreover, the acquisitions carried out at CARElab with ASD children are exposed.

Then, in Chapter 5, overall results of both gesture recognition algorithm and acquisitions are analyzed and discussed.

Finally, in Chapter 6 final conclusions are pointed out and several directions for future improvements and research challenges are outlined.

Chapter 2

State of the Art

2.1 Autism

Autism Spectrum Disorders (ASDs) are a group of neuropsychiatric disorders characterized by deficits in social communication as well as by the presence of restricted interests and stereotyped and repetitive behaviors [18]. There is no cure, the causes are still unknown and symptoms vary from patient to patient.

The prevalence of ASD has been increasing in the past two decades: it is now 1 in 54 children based on 2016 data, up from 1 in 60 in 2014 [19]. Increased ASD screening frequency in children and adults, better diagnostic criteria and more accurate behavioral and neuro-psychological scales may all have also contributed to the steady rise in the prevalence of ASD. The spectrum of symptoms can be very broad: deficit in sharing emotions or affect, failure to response to external stimuli, difficulties in understanding situations and relationships or to suit to different social contexts. Stereotyped movements or behaviours, difficulties in the adherence to routine and hyper- or hypo-reactivity and interest in unusual external stimuli may often be present too. Those symptoms can be accompanied by intellectual and language impairment or not [20].

ASD is diagnosed when a patient demonstrates at least three symptoms in the domain of social communication and at least two symptoms of restricted interests/repetitive behaviors. Assessment instruments include parent/caregiver interviews, patient interviews, direct observation of patients and detailed clinical assessments [21].

Heterogeneities in etiology, phenotype and outcome are hallmarks of ASD. The subsequent clinical variety shows different levels of deficits or impairments in behavioral features and communicative functioning [20]. Heterogeneity is mainly due to environmental factors, but also to gender, multiplicity of genes involved and genetic variability. Gender distribution seems to have a role in ASD since there is a prevalence of 3 males diagnosed to every 1 female [22] and genetic architecture in ASD varies substantially from single mutation being enough to cause ASD, to an accumulation of over one thousand low-risk alleles [23]. Even comorbidities highlight and complicate the heterogeneity of ASD. Comorbid psychopathologies include anxiety, depression, Attention Deficit Hyperactivity Disorder (ADHD) and intellectual disability. Moreover, medical comorbidities include seizures, sleep difficulties, gastrointestinal disorders, mitochondrial dysfunction and immune system abnormalities [20].

Few studies have been conducted to understand how symptom description, interpretation, and acceptance of ASD may vary along different population, but evidence has suggested that culture influences the diagnostic process, intervention services provided and the outcome for an individual with autism. Therefore, treatment options have to be set accordingly to a specific goal [24]. Other factors contributing to the difficulties in identifying efficacious treatments include small sample sizes, the lack of significantly impaired study participants, highly variable study samples, which reduce the potential effect size of an intervention, and the use of outcome measures that are not uniformly adopted. Behavioral interventions undertaken early in life, using an intensive delivery format, are considered the current gold-standard treatment for behavioral symptoms associated with ASD. Alternatively, only two pharmaceuticals were approved by the US Food and Drug Administration (FDA), risperidone and aripiprazole [20]. Recently, new therapies involve a variety of equipment to improve cognitive skills and social interaction of ASD children such as tablets for different games, computer used with joystick and mouse, mobiles and socially assistive robots. All of these interventions aim to increase the children concentration and improve features like eye contact, eye blink rate, response time, task repetition, proximity with peers in terms of distance, joint attention, turn taking and communication [5].

2.2 Robot Therapy

The challenges faced by people with autism when interacting with others are characterized by confusion, fear or basic misunderstanding of emotions. They have difficulties using and understanding verbal and non-verbal communication, recognizing and properly reacting to other people's feelings [3].

In contrast, autistic people cope well with rule based, predictable systems such as computers. Recent developments have shown the advantages of using humanoid robots for psycho-educational therapy. Children with autism feel more comfortable around such robots than humans, who may be perceived as hard to understand and sometimes even frightening. Thus, most studies are based on remote controlled Human-Robot Interaction (HRI) [3]. Different structured scenarios based on activity or a physical play between a socially assistive robot and ASD children have produced results in the communication and social behavior of the children [5].

Robot-Assisted Therapy (RAT) enables embodied interactions, such as increasing engagement and attention and decreasing social anxiety. During a child-robot interaction, robots can simultaneously provide social cues while maintaining simplicity and predictability. The robots used in RAT differ in their appearances, ranging from mobile platforms to humanoid robots [4]. Furthermore, recent researches report how humanoid robots can help to increase bodily awareness of children with autism [25]. Despite the promising results of RAT, most of the studies are exploratory and have methodological limitations, such as a low number of participants or failures to comply with therapy protocols [4]. Robots for autism therapy can play different roles: demonstrators and guides of the interaction, toys responding to the child, mediators of the social behavior between the child and others. To this purpose, a robot can verbally ask the child to perform certain behaviors, assist the child in predetermined play scenarios, or move autonomously in order to let the child start an imitation task or free-play interactions by himself/herself. More frequently, a therapist

or teacher guides the child through robotic interactions, for instance by asking the child to touch the robot or to imitate the robot's behavior [26]. The different roles assumed by the robot during RAT are correlated with specific control paradigms: Wizard of Oz (WoZ), full autonomy and supervised autonomy (Figure 2.1). Most RAT studies are limited to

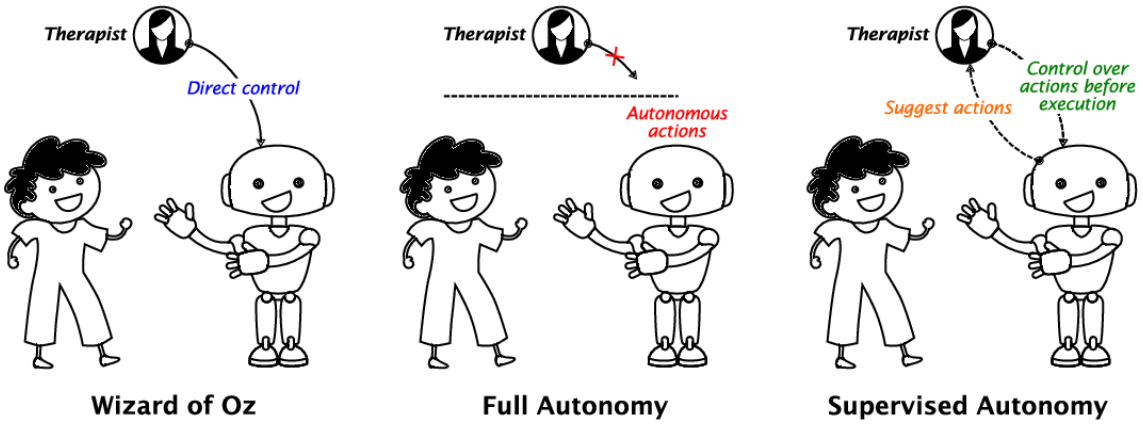

Figure 2.1: Robot control paradigms [4].

the WoZ technique in which robots are remotely controlled by a human operator, without the knowledge of the child. Increase the level of robot autonomy in RAT research is important in order to decrease the human workload and to deliver consistent therapies. In the full autonomy control paradigm the robot makes decisions and adapts its actions to any situation by itself. Since the robot's action policies cannot be perfect and its behaviours must be compliant with the therapeutic goals, interaction context and state of the child, this paradigm is not feasible and can raise some critical ethical concerns. Therefore, a "supervised autonomy" in which the robot works independently toward achieving given therapeutic goals under a supervisor's guidance is suitable and known as RET. The robot is provided with the information necessary to select its next actions within well-defined constraints under the supervision of a therapist [27]. For instance, in the semi-autonomous implementation of Kaspar robot by Zaraki et al. [28], the robot played four individual games with two children and a joint game with a pair of children while a researcher was sitting next to the robot to facilitate the interaction by evaluating the robot's behaviour and giving the final permission for the robot to display the behaviours recommended by the system.

As observed by Sial et al. in their work [5], a collaborative approach based on interactive games between a robot and ASD children has produced positive results in terms of therapeutic outcomes such as social interaction, communication, joint attention and turn taking. Studies have shown that motor impairments are a prominent comorbidity within the ASD phenotype [6], even though they are not currently included in the diagnostic criteria of autism. McAuliffe et al. [7] explored how these altered motor ability might be linked to an atypical skills learning. In point of that, their results supports the hypothesis that a poor imitative gestural learning can impact social and motor development, since learning via imitation is a prime method by which humans acquire skills. As a consequence, imitation skill training should be included in intervention programs [2]. In this context, robotic technologies have been shown to be valuable tools for ASD therapies

[29, 30]. Thanks to gesture classification algorithms, robots are able to provide feedback with the aim of increasing the engagement of children in the gesture imitation programs. Storytelling is one of the systematic intervention for ASD children to learn gestures via imitation. Storytelling has demonstrated promising results in improving social perception, social and cognitive skills and interactions [30, 31, 2]. Many research studies suggested that robots can help in storytelling activity [30], but it's important to underline that this kind of therapy is not standardized. According to [31], the main goals are:

- provide social support, establishing confidence and reducing stress;
- create a pleasant environment (facilitate social play);
- establish a dynamic model of social interaction (social interaction peers);
- strengthen motivation, increasing personal initiative;
- improve communication;
- use non-verbal communication (improve eye contact, facial expressions and gestures);
- engage in play, developing imitation;
- develop empathy;
- support active learning, encouraging participation;
- integrate targeted behaviors into learning.

In the context of gesture recognition and imitation training, following the structure proposed by Duarte et al. [2], the story follows a linear structure with an introduction, a mid point and a conclusion. While in the first part the scenario is introduced, in the mid point the gesture is shown (by the therapist or by the robot) and imitation training starts. In the conclusion, a reinforcement is given depending on the performance of the child (e.g. waving back if the child performed well) [2]. This reinforcement could be triggered automatically thanks to a gesture recognition system able to evaluate the performance of the child by means of gesture classification. During the session, the therapist selects a gesture or a skill to be learned and also an appropriate story. The therapist controls the transitions of each part of the therapy: when clarification is needed (e.g the child is distracted by other stimuli), the story can be played back. In this way, it is possible to improve imaginative abilities and social competences through gestures representing social skills such as: pointing, showing, giving, clapping, waving etc. [2]. However, a more extensive intervention could be necessary in order to facilitate the transfer of skills to ADL [31]. Moreover, it is important to take into account that children with ASD can perform the gestures with certain shades/nuances that make it challenging for gesture recognition systems to recognize the movement and it can also be hard to evaluate child's performance. For instance, it is common that the waving/goodbye gesture is performed with the child's palm of the hand facing himself instead of the person he/she is waving to or the child raises the arm without waving: in these cases his performance has to be considered very positive, even if the gesture was not correctly executed [2].

2.3 Gesture Recognition System

The word "gesture" refers to any non-verbal communication, intended to deliver a specific message. In the field of gesture recognition, a gesture is described as any physical movement with a dynamics executed over time that can be interpreted by a sensor. The general definition of gesture recognition is the ability of a computer to understand gestures and to execute commands based on those gestures. In this context, recognition can follow four different steps: Data Acquisition, Data Processing, Feature Extraction and Classification [2]. A focus on the way Online Recognition can be implemented can be carried out.

2.3.1 Data Acquisition

The first step consists in data extraction, which converts the physical gesture to numerical data. Online recognition tasks depend on data acquisitions instruments. In vision-based capturing systems, the gesture is identified by a camera and the main advantage is that they allow the natural execution of the subjects' movement. On the other hand, the main drawbacks are the complexity of processing and the camera field. Since children with autism are particularly sensible to the touch [8], wearable systems would be unfeasible, even if they are fast, robust and receive information directly from users' movements. Microsoft Kinect cameras are the most used tools (Figure 2.2). These tools are a combination of an RGB camera and a depth sensor. In the first version, the depth sensor used an infrared technology while in the second version it uses time-of-flight technology to create 3D images. Due to the 3D pose estimation algorithms intrinsic to the Kinect camera, it is able to estimate the joint 3D positions of two people in front of the camera, called skeleton points (Figure 2.3).

Figure 2.2: Kinect Joint Map.

Therefore, the data extracted from a Kinect camera can be of three types: arrays of skeleton points [2, 32, 9, 33, 34], RGB-Depth images [3] or a combination of both [35, 36, 11].

Datasets

When comparing results in literature, the type of datasets used must be taken into account in sight of the development of a classification system. Du et al. [35] evaluate their model on both Berkeley Multimodal Human Action Dataset (Berkeley MHAD) and ChaLearn dataset. Berkeley MHAD is characterized by 11 actions performed by 12 subjects formed by movements in both upper and lower extremities or by movements with high dynamics in upper or lower extremities. Furthermore, the subjects were given instructions on what action to perform; however no specific details were given on how the action should be

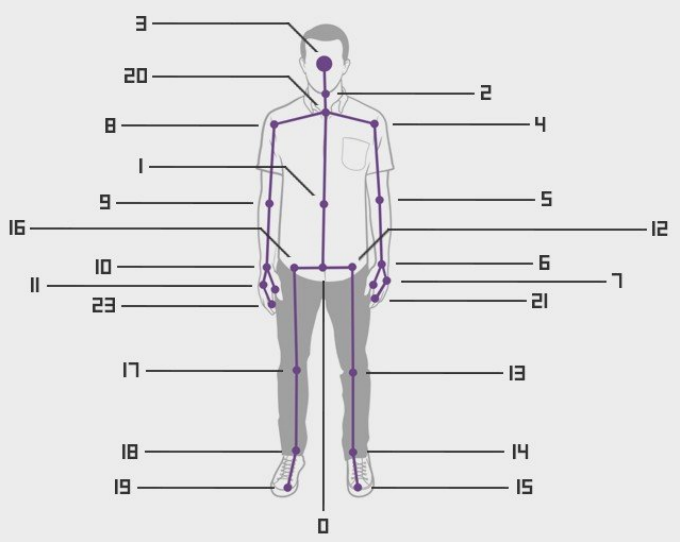

Figure 2.3: Kinect Joint Map

executed. On the other hand, ChaLearn gesture recognition dataset is made of 27 people performing 20 Italian gestures. Hou et al. [11] experimented their work on three public datasets: MSRC-12 Kinect Gesture Dataset contains 12 gestures involving all the body parts performed by 30 subjects previously coached on how to perform each gesture, Gaming3D Dataset is formed by 10 subjects performing 20 gaming actions while UTD-MHAD dataset is characterized by 27 actions performed by eight subjects (four females and four males) with each subject performing each action four times. Mathe et al. [37] evaluate their approach on a real-life dataset of 10 users with 5 samples per 10 gestures, which involved the upper body part only. In this case, the subjects were provided only with an intuitive description of the way gestures should have been performed. Zhang et al. [38] experimented their recognition method on two public datasets: ChaLearn LAP largescale isolated gesture dataset (IsoGD) and Sheffield Kinect Gesture dataset (SKIG). IsoGD is made of 249 kinds of gestures performed by 21 individuals, while SKIG contains 10 categories of hand gestures and all gestures are performed by 6 individuals with 3 kinds of hand postures under 2 illumination conditions and 3 backgrounds. Wang et al. [36] used three datasets: NTU RGB+D dataset, SBU Interaction dataset and ChaLearn Gesture Recognition dataset. NTU RGB+D dataset is made of 60 different all body-actions performed by 40 different subjects, SBU Interaction dataset includes 8 activities performed by 714 participants and ChaLearn Gesture Recognition dataset contains 20 Italian gestures performed by 27 different people. In the end, Pham et al. [9] exploited MSR Action 3D dataset and KARD dataset. MSR action dataset contains 20 different Activities of Daily Living (ADL) performed by 10 subjects for three times while KARD dataset contains 18 ADL performed by 10 different subjects. In testing their model, they divided both datasets in 3 subsets of 8 actions each. Table 2.1 summarizes the most used datasets.

2.3.2 Data Processing

Regarding data processing, different techniques are widespread used, being the most common: the normalization of the skeleton's size and video pre-segmentation. The first one

Table 2.1: Datasets' Comparison.

Cite	Dataset	gestures	Body parts
[35]	Berkeley MHAD	11	all
	ChaLearn	20	all
[11]	MSRC-12 Kinect Gesture Dataset	12	all
	Gaming3D Dataset	20	all
	UTD-MHAD	20	all (but mainly upper body)
[37]	real-life dataset	10	upper body
[38]	ChaLearn IsoGD	249	all
	SKIG	10	hand
[36]	NTU RGB+D	60	all
	SBU Interaction	8	all
	ChaLearn	20	all
[9]	MSR	20	all
	KARD	18	all
[33]	MSR	20	all
	KARD	18	all
	NTU RGB+D	60	all
[34]	MSR	20	all
	KARD	18	all
	NTU RGB+D	60	all
	SBU Interaction	8	all

acts on skeleton's coordinates: since different users have different physical characteristics (height, limb length) and might be standing at different distances from the capturing device, the coordinates are scaled with respect to the skeleton height (in order to compare skeletons with the same size) and centered with respect to a reference point (e.g. hip joint or neck joint) to set it as the origin [2, 3]. On the other hand, video pre-segmentation allows to isolate the gesture by manually trimming the frames that do not contain the movement [2, 32].

2.3.3 Features

The third step characterizes gestures, recurring to specific features describing the kinematics of actions executed. Marinoiu et al. [3] exploited both 2D and 3D pose features (Figure 2.4). The first ones were characterized by 2D body joints locations. Even though good accuracy and speed were obtained, 2D information might be insufficient for actions' interpretation, as the depth information could be crucial. That is why, subsequently, 3D human skeletons were extracted from Kinect disregarding the temporal information. Papadakis et al. [32] used signal images, based on the creation of 2D images by concatenation of 1D signals which captured the 3D motion of human skeletal joints over space and time. Different transformations were applied for the creation of "activity" images (e.g. Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST)) (Figure 2.5). Wang et al. [36] chose "Joint Trajectory

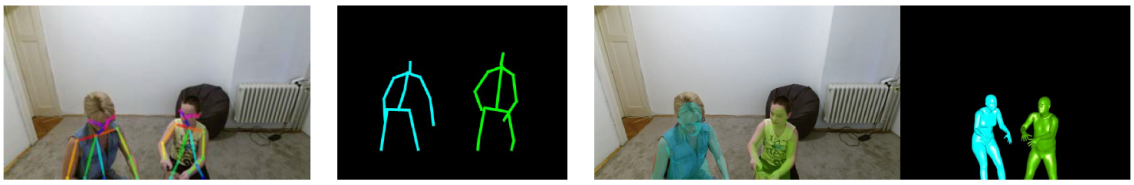

Figure 2.4: Examples of 2D and 3D pose reconstruction. From left to right: 2D joint position estimates, 3D pose estimation, projection of the inferred shape model overlaid on the original image and inferred 3D shape model [3].

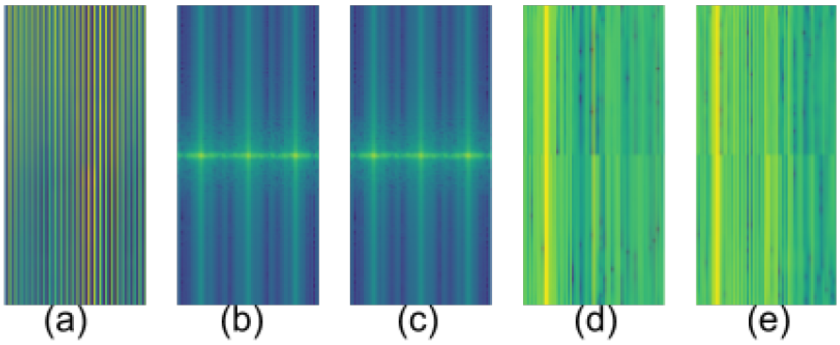

Figure 2.5: (a) A signal image; activity image resulting upon (b) DFT; (c) FFT; (d) DCT; (e); DST. [32].

Maps" in order to encode the motion information into texture patterns by setting saturation and brightness; similarly, Hou et al.[11] encoded the temporal variation changing the hue channel in the construction of "skeleton optical spectra" (Figure 2.6). On the other hand,

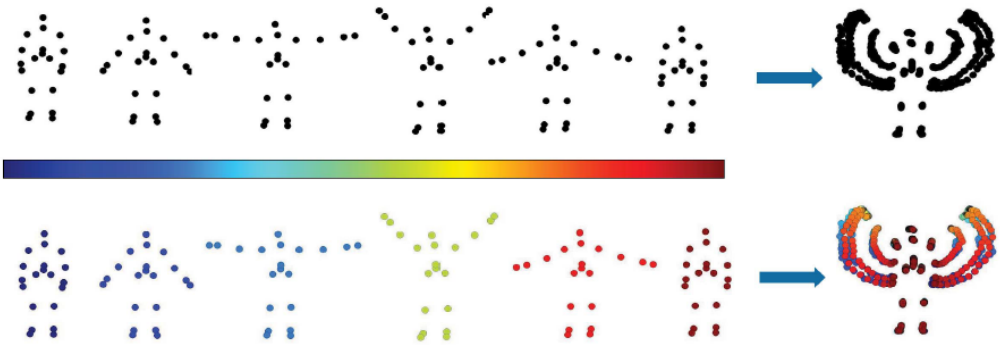

Figure 2.6: Joints at different time-stamps have different colors reflecting the temporal order. [11].

Du et al. [35] and Pham et al. [9] employed coordinate projections on three orthogonal planes represented as three RGB components (Figure 2.7).

Once the features are selected, further processing can be carried out. As a further step for characterizing gestures, Pham et al. [34] proposed a method to enhance the local patterns of their RGB representations through an Adaptive Histogram Equalization (AHE), which is able to enhance the contrast of an RGB representation locally. As an Histogram Equalization technique, AHE adjusts the gray level of an image according to its probability

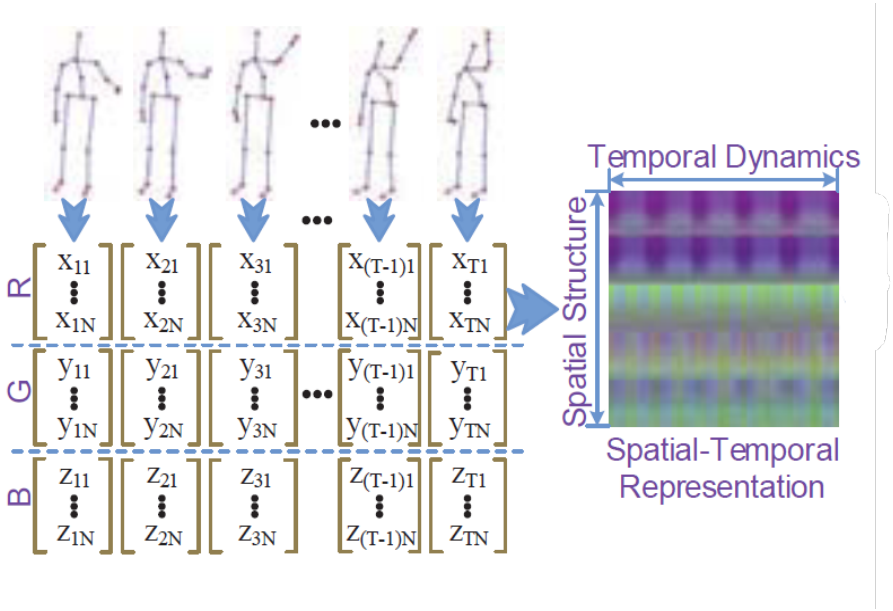

Figure 2.7: Three components of all skeleton joints in each frame are separately concatenated by their physical connections. After arranging the representations of all frames in chronological order, the generated matrix is quantified and normalized into an image [35].

distribution function and enlarges the dynamic range of the gray distribution to improve visual effects [39]. In fact, to enhance the image contrast, Histogram Equalization spreads out the most frequent pixel intensity values or stretches out the intensity range of the image (Figure 2.8). By accomplishing this, image's areas with lower contrast gains higher contrast.

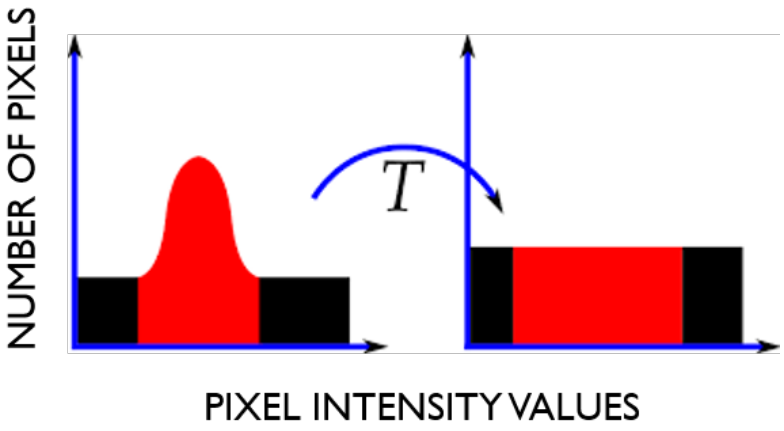

Figure 2.8: Histograms of an image before and after equalization.

2.3.4 Classification

The last step is gesture classification whose aim is to interpret signals to label human movements. Gestures could be static or dynamic. The former are time independent and easier to classify while the latter need more sophisticated techniques to face temporal evolution.

Machine learning is a frequently used tool in this field since it provides the systems with the capacity to automatically learn and improve from past experience without being explicitly programmed [40]. Machine learning algorithms for gesture recognition can be divided in supervised and unsupervised learning algorithms.

Supervised and Unsupervised Learning Algorithms

Supervised learning algorithms learn from labeled data to create models able to classify new data. The most used in the gesture recognition field are K-Nearest Neighbors (KNN), Hidden Markov Model (HMM), Support Vector Machine (SVM) and Artificial Neural Network (ANN)s. Lai et al. [41] used KNN to recognize hand gestures in real-time using Kinect camera. They create a dataset of skeleton sequences for 20 individuals performing 8 simple hand gestures useful for human-computer interaction. Each gesture was repeated 5 times by each individual and recorded over 1 second (30 frames). The method had an accuracy of 97.2% but it's sensitive to temporal misalignment and more storage- and computation-heavy than their former work based on a log-covariance method [42]. Anuj et al. [43] based their approach on HMM and adaptive thresholds to classify a set of gestures to control the basic operations in a PowerPoint presentation. Their test dataset included 5 subjects performing 5 gestures 16 times – 8 times correctly and 8 times incorrectly. They chose HMMs to handle the time series data and classify sequences. They created a discrete HMM for each dynamic gesture and fed the stream simultaneously to all HMMs. Each HMM, in turn, returned a likelihood for match. Compared to a single HMM for all gestures, multiple HMMs better preserve the discriminating features of every gesture and have better accuracy. The precision obtained was high (96.48%), but this approach exploited different features for different kinds of movements and analyzed specific frames depending on the gesture to be recognized. Gu et al. [44] used an HMM for each gesture on sets of 3D joints: 5 gestures are defined for the experiments (come, go, wave, rise up and sit down). Since HMM has already become a general method to modeling speech signals, they applied it in the field of gesture recognition because of the similarities between temporal gesture signal and speech signal. Their method reached an accuracy of 85.0% with low detection speed. The disadvantages of using HMMs are the need for an *a priori* notion of the model topology and, as with any statistical technique, large amounts of training data [45]. Bhattacharya et al.'s [46] approach used SVM on 3D skeletal joint coordinates of edited data stream first (the starting and ending frames of each gesture were marked by a human observer) with an accuracy of 99.97%. Then they extended those techniques to detect and classify a gesture in an unedited stream which also captures random movements. This recognition was done offline and reached an accuracy of 83.33%. The gesture vocabulary was based on aircraft marshaling gestures used in the military air force and included 8 actions and tested on 3 subjects.

Unsupervised learning algorithms are based on non-labeled data and they learn how

to classify autonomously; K-means and Artificial Neural Network (ANN) are examples of unsupervised learning algorithms. Gani et al. [47] in order to differentiate signer's hands used a K-means clustering algorithm to partition pixels into two groups corresponding to each signer's hands. Every gesture in testing data set is then compared against each gesture in training data set by using Euclidean distance. Two data sets have been created, corresponding to training and testing data set captured from 4 different signers using both their hands. They reached an accuracy of 91%. Maharani et al. [48] compared SVM and K-means methods showing that Multiclass SVM (99.15% accuracy) performs better than K-Means clustering method (77.42% accuracy). The testing was done on 6 peoples, and each person was tested 180 times with four gestures (forward, right, left, and stop), three distances (2m, 3m, 4m), and three slopes positions (45°, 0°, -450°).

In recent years, among ANN, Convolutional Neural Network (CNN) has become a crucial algorithm in image classification field for gesture recognition because of its fast and robust classification ability. It has the capacity of feature learning without the need of extracting features manually and it can train unprocessed images and generate feature extraction classifier automatically [49]. CNN is a Deep Learning algorithm which can take in an input image, enhance various patterns in the image and be able to discriminate one from the other. The architecture of a CNN was inspired by the organization of the Visual Cortex. Each neuron responds to stimuli only in a restricted region of the visual field: the receptive field. In other words, the inputs of hidden units in layer n are from a subset of units in layer $n-1$ (Figure 2.9). This particular architecture is characterized by local spatial

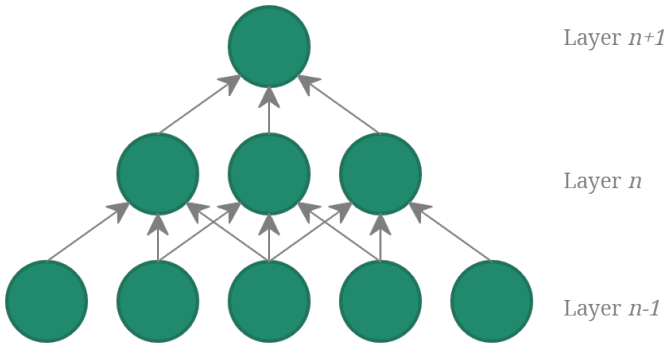

Figure 2.9: CNN wiring exploiting spatially-local correlation.

coherence which allows the net to learn features without training millions of parameters, since some of them are shared. In this way, the extraction of relevant information occurs at low computational cost. In the context of gesture recognition, there are several types of CNNs such as ResNets. ResNets can have variable sizes, depending on how big each of the stages of the model are, and how many of them it has. Each stage is composed of a number of residual blocks, each of them characterized by weight layers, as shown in Figure 2.10. In Figure 2.10, X is the predicted label that must be equal to the true label. The residual function $\mathcal{R}(\mathrm{x})$ (error function) will compute and produce the residual of the model (error measure) to match the predicted label with the true label. Thus, a residual block learns a mapping function $y = \mathcal{R}(\mathrm{x}) + \mathcal{S}(\mathrm{x})$ where $\mathcal{R}(\mathrm{x})$ is a set of weight layers and $\mathcal{S}(\mathrm{x})$ is the shortcut connection's function. These shortcut connections skip the training of one or more

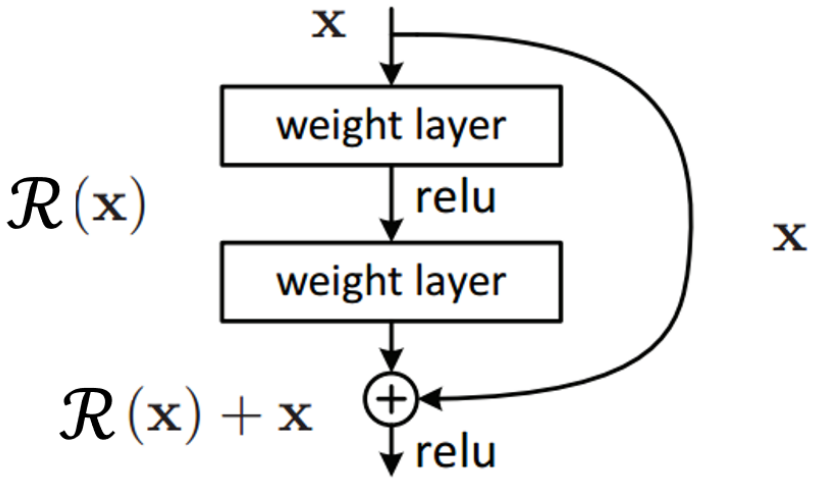

Figure 2.10: A single residual block of ResNet. More residual blocks constitute one of the stages of ResNets.

weight layers. During the computation of the gradient, the skip connection path allows it to effectively reach initial layers, skipping the middle ones. In this way, the gradient does not back propagate layer by layer thus solving the vanishing gradient problem and allowing a deeper network. The main advantage of deeper models is the ability of performing more convolutions extracting with more precision the relevant features. Hence, ResNet is able to build a deeper neural network without the risk of degradation in performance. Moreover, ResNets' extracted features in lower layers are raw and elementary, while those in upper layers are high-level abstract features as combinations of the first ones. In fact, the output from the first hidden layer extracts certain attributes from the input, so the information content from the first hidden layer is much richer than the input itself. For this reason, the second hidden layer needs to have a larger number of filters to properly extract the now richer features. Because of the increasing in the number of filters, the feature map's size has to be decreased to preserve time complexity per stage [50]. Downsampling is achieved by a convolutional layer with a specific stride rather than by a max pooling layer. When pooling is replaced by an additional convolution layer with stride, performance stabilizes and even improves on the base model [51]. Moreover, pooling is a fixed operation while convolution can be learned. For all these reasons, input/output dimensions can be different and two main types of residual blocks are used in a ResNet: the Identity block – the case where the input activation has the same dimension as the output activation ($\mathcal{S}(\mathrm{x}) = id(\mathrm{x}) = x$) and the Convolutional block – when the input and output dimensions don't match up and there is a Conv2D layer in the shortcut link ($\mathcal{S}(\mathrm{x}) = Conv(\mathrm{x})$).

Several types of CNN are used in the field of recognition tasks. Du et al. and Papadakis et al. [35, 32] used skeletal information to create images capturing the motion of joints in the 3D space and to feed a CNN. Du et al. evaluate their model on both Berkeley Multimodal Human Action Dataset (Berkeley MHAD) reaching 100% accuracy and ChaLearn gesture recognition dataset with a precision of 91%. Hou et al. [11] found an effective method to

encode the spatio-temporal information of a skeleton sequence into color texture images, referred to as Skeleton Optical Spectra (SOS), and employs CNNs to learn the features for action recognition. Experiments were conducted on three public datasets. With MSRC-12 Kinect Gesture Dataset the method reached an accuracy of 94.27% while with Gaming 3D Dataset it reached an accuracy of 95.45%. For the challenging UTD-MHAD dataset the accuracy was 86.97%. Mathe et al. [37] used a CNN on raw skeletal data (3D coordinates) to classify arm gestures and evaluate this approach on a real-life dataset of 10 users. Even though the subjects were only provided with an intuitive description of the way gestures should have been performed, their model reached an accuracy of 90%. Zhang et al. [38] used 3D CNNs and convolutional Long Short Term Memory (LSTM)s. Two public datasets are used to evaluate the performance of their proposed method: ChaLearn LAP largescale isolated gesture dataset (IsoGD) and Sheffield Kinect Gesture dataset (SKIG). In addition to RGB and depth data, they also used optical flow data to improve the prediction accuracy. The method reached an accuracy of 62.14% and 99.53% with the two datasets, respectively. Wang et al. [52] proposed two-stream Recurrent Neural Network (RNN) architecture to model both temporal dynamics and spatial configurations for skeleton-based action recognition. Their model was evaluated on three datasets: with NTU RGB+D dataset they reached a cross-subject accuracy of 71.3%, with SBU Interaction dataset they got an accuracy of 94.8% and with ChaLearn Gesture Recognition dataset they got a precision of 91.7%. Pham et al. [9] used a CNN ResNet on RGB images in which skeleton sequences transformed into 3D arrays were exploited. They evaluate their model on two datasets: MSR Action 3D dataset and KARD dataset. More specifically, they divided both datasets in 3 subsets of 8 actions each obtaining, with their best net configuration, accuracies of 99.4%, 99%, 100% and 100%, 100%, 100%, respectively.

2.3.5 Online Recognition

Tipically, in the context of gesture recognition that relies on vision-based systems (e.g. Kinect cameras), a camera records human bodies and the system extracts features (e.g. 3D coordinates) from the individual frames of the recording. In their work, Luzhnica et al. [53] use sliding windows as basic unit for real-time classification, i.e. data windows of fixed size (number of frames) that constitute snapshots of the continuous data stream. Features are computed per window, which typically has two configuration parameters: *size* and *step*. For parameter selection, they cross validated the data with several window sizes (140, 160, 180, 200 number of frames, where 1 second contains 85-87 frames). They also used steps of 20, 30, 40 and 50 frames and again used cross validation to select a value for this parameter. Molchanov et al. [54] worked on offline and online hand gesture recognition. To detect the presence of a *no gesture* they compare the highest current class conditional probability output by the net to a threshold $\tau \in [0,1]$. When the detection threshold is exceeded, a classification label is assigned to the most probable class. Baldissera et al. [55] implemented online recognition using a window of the 16 most recent *tables* (each containing a fixed number of frames) updated with each new acquired framework. Their proposed model evaluates the window in question and saves the softmax probabilities of this result in a *buffer* containing the latest N predictions. If the average of any class in the buffer exceeds a limit of network trust (C_{th}), the algorithm then identifies that this

gesture took place. To prevent the same gesture to be classified multiple times, after the assignment of a label the algorithm has a time of *cooldown*, in which it is silenced, even if some class exceeds the confidence limit. This way, the window that stores the latest frames is renewed preventing the same sequence of frames that generated a classification of a gesture to be analyzed again.

Chapter 3

Methods

The goal of this thesis project is to find a proper method to recognize gestures inside robot therapies for children with ASD. In this way, the robot could react properly to children's movements when interacting with them and support therapist's work in the Therapist-Robot-Child triad therapy (Figure 3.1). Experiments of this kind of treatment take place at CARElab (Computer Assisted Rehabilitation) in Fondazione Don Gnocchi (Milan) with IOGIOCO robot therapy. The gesture recognition algorithm is part of the therapy protocol and its workflow was: Data Acquisition, Data Processing, Pose Features, Classification and Online Recognition (Figure 3.2). In this chapter IOGIOCO protocol, algorithm workflow and algorithm integration in the therapy are presented.

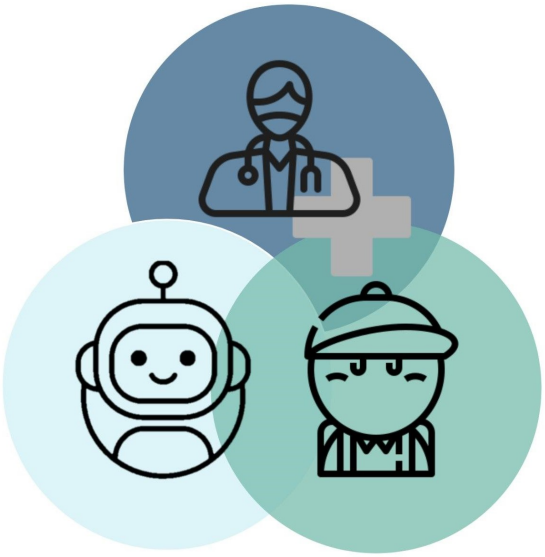

Figure 3.1: Therapist−robot−child triad.

3.1 IOGIOCO Robot Therapy

IOGIOCO aims to empower significant transitive and intransitive gestures in children with ASD, thanks to interactive mirroring games with the humanoid robot NAO. NAO robot is 0.57 m high and 0.28 m wide with 25 degrees of freedom of manipulation. It was designed

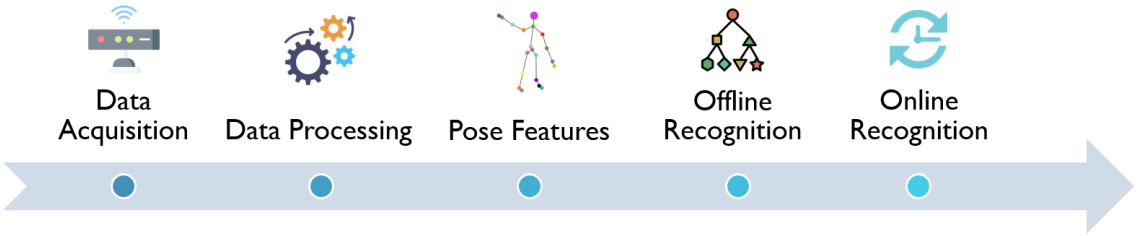

Figure 3.2: Workflow of the proposed algorithm.

to answer to a variety of inputs, offering different set of tasks: from walking to grabbing objects, or even stand-up by itself after a fall [56]. IOGIOCO protocol includes 5 training

Figure 3.3: NAO Robot.

phases/levels and one final evaluation. The sessions have an approximate duration of 10−20 minutes and are executed weekly. The different levels are:

- Level 1: Familiarization;
- Level 2: Pure mirroring;
- Level 3: Introduction of selected specific gestures;
- Level 4: Integration of learnt gestures in child's Activities of Daily Living (ADL) (tasks of every-day life);
- Level 5: Further generalization of the proposed gestures.

The first two phases have the purpose of encouraging the child to adapt to the setting and to the proposed activities. The following three phases involve the training of selected communicative gestures within a Robot-Child-Therapist interaction. One of the modalities in the triad is the Robot Coach (RC). The robot shows the action to be performed, then therapist and child respectively have to repeat the gesture. For each person, the robot gives some visual and/or sound feedback depending on the performance. In Therapist Coach (TC) modality, the therapist shows the action and NAO mirrors it offline. Then the child has to perform the action introduced by the therapist, while NAO is mirroring him/her. In this modality, the feedback should be given by the therapist.

3.1.1 Gestures

The gestures selected are meaningful from the communicative point of view and part of ADL. The 19 gestures of the protocol are: *tall*, *angry*, *listening*, *waiting*, *kissing*, *short*, *giving*, *where*, *hungry*, *me*, *peekaboo*, *happy*, *yes*, *no*, *big*, *hello*, *little*, *pointing* and *coming* (Figure 3.4). All gestures are inserted in a narrative context consisting of short sentences

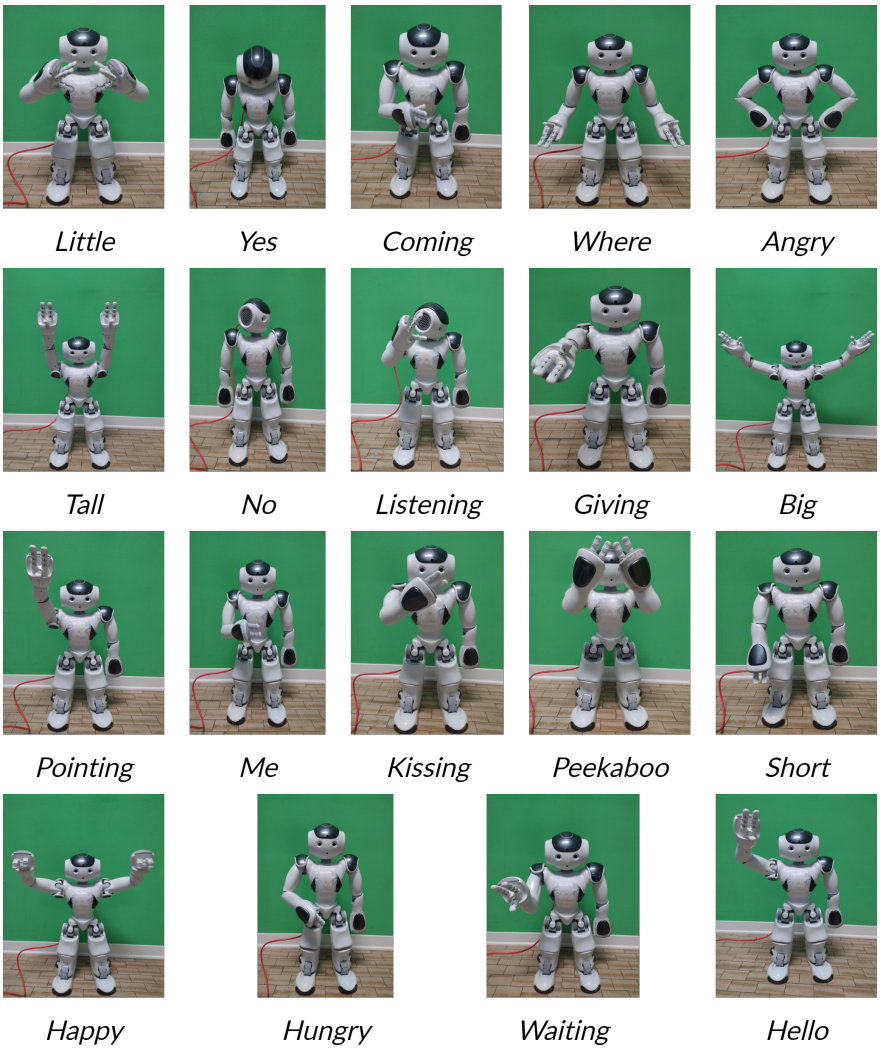

Figure 3.4: All gestures performed by NAO.

organized in small dialogues, for example, "I am hungry". The quantity and the different lengths of movements used in IOGIOCO protocol make the learning process of a recognition algorithm challenging. This recognition algorithm is part of the protocol from Level 3 on.

3.1.2 IOGIOCO Level 3

Level 3 begins only when the child is fully familiarized with the proposed new settings and with NAO. This phase is the first level in which specific intransitive gestures with communicative purpose are introduced. To start this phase, the therapist selects a specific gesture from a screen panel interface, which allows to define which gestures to perform inside a single session. Moreover, for each repetition it is possible to indicate the modality. Depending on the mode of interaction chosen (RC or TC), the control is on the robot or the therapist. RC modality has been implemented with the new gesture recognition algorithm. In RC modality, once the gesture is selected, the activity is structured as shown in Figure 3.5. The first "actor" in RC is the therapist, who supervises the therapy's level

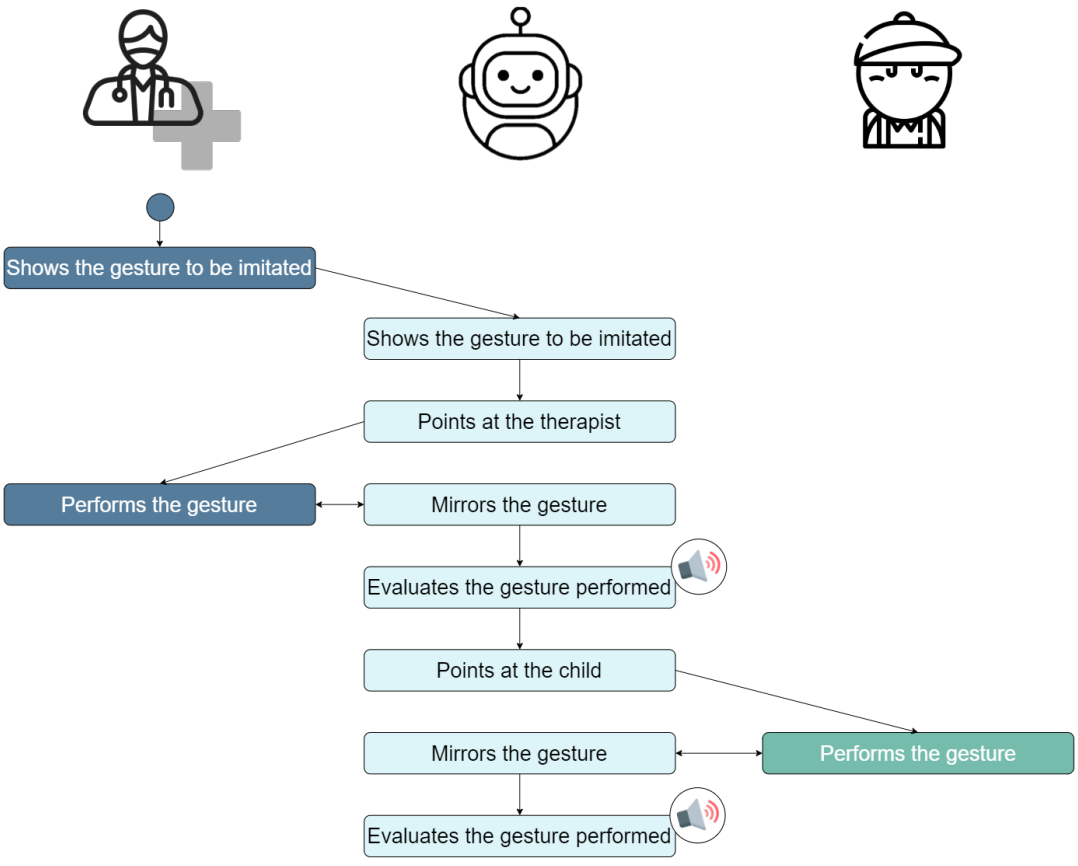

Figure 3.5: IOGIOCO therapy protocol: Level 3 Robot Coach's phases.

deciding the gesture to be performed and taught to the child. Then, the robot performs the gesture to be imitated both by the therapist and the child and points at the therapist first. The moment the therapist starts performing the action, the algorithm begins the evaluation and, after a specific temporal window (about 10 seconds), the robot gives a positive or negative sound feedback saying "Well done!" or "Come on, let's do it again!"

respectively. After a fixed amount of time, the robot points at the child and the classifier starts its evaluation again.

3.2 Data Acquisition

In order to acquire movement data, the tool used was a Microsoft Kinect v2 camera, now on referred as Kinect. It is able to capture keypoints (skeleton 3D coordinates) and jointpoints (skeleton 2D coordinates in relation to the camera image). Among these, keypoints of each subject were selected to be analysed, since they allow a spatial analysis of each gesture. Figure 3.6 represents Kinect Joint Map.

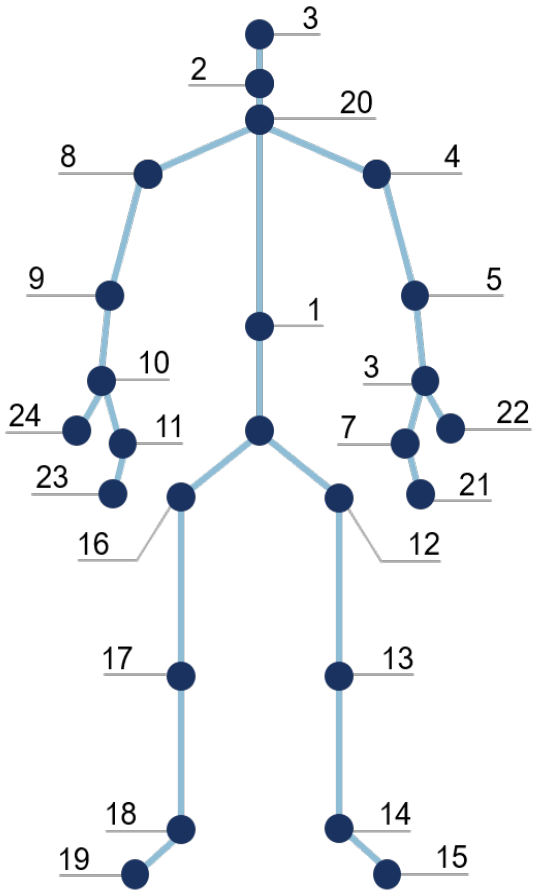

Figure 3.6: Joint Map

Three different datasets were acquired and used for the development of the algorithm of gesture recognition.

3.2.1 Subsampled Healthy Dataset

The first dataset was constituted by a subsample of gestures: *small*, *hello*, *pointing*, *come* and *yes*. 18 healthy adult subjects performed these gestures, captured by a Kinect. Each gesture was repeated 14 times with a total number of samples of 1260 (18 healthy adults $\times$ 5 gestures $\times$ 14 times). Each sample contained a single repetition of a single gesture,

Table 3.1: Joints' Dictionary.

Joint Type	Keypoint	Joint Type	Keypoint
Hip Center	0	Left Knee	13
Middle Trunk	1	Left Ankle	14
Neck	2	Left Foot	15
Head	3	Right Hip	16
Left Shoulder	4	Right Knee	17
Left Elbow	5	Right Ankle	18
Left Wrist	6	Right Foot	19
Left Hand	7	Shoulder Center	20
Right Shoulder	8	Left Hand Tip	21
Right Elbow center	9	Left Thumb	22
Right Wrist	10	Right Hand Tip	23
Right Hand	11	Right Thumb	24
Left Hip	12		

so no segmentation was required. This dataset was used for a first implementation of the algorithm.

3.2.2 Healthy Dataset

Once the algorithm was implemented, a more comprehensive dataset made of 18 healthy subjects, 9 adults and 9 children, was exploited. In each acquisition, a "therapist" and a "child" were asked to perform the 19 therapy protocol's gestures captured by a Kinect: *tall*, *angry*, *listening*, *waiting*, *kissing*, *short*, *giving*, *where*, *hungry*, *me*, *peekaboo*, *happy*, *yes*, *no*, *big*, *hello*, *little*, *pointing* and *coming*. Unlike the Subsampled Healthy Dataset, each sample contained all gestures executed by a subject. Therefore, segmentation was required. Each file was split in several isolated actions according to a starting and an ending frame, defined manually. The gesture was considered as a combination of movements that could be repeated different number of times (multiple repetitions per gesture were considered), since this execution difference was verified in the several subjects. Then, to balance the dataset, gestures with a low number of samples (less than 23) were neglected. In the end, actions were reduced from 19 to 14: *tall*, *angry*, *listening*, *waiting*, *kissing*, *short*, *giving*, *where*, *hungry*, *big*, *hi*, *little*, *pointing* and *coming*. The total number of files obtained was 367.

3.2.3 Expanded Dataset

The aforementioned Healthy Dataset was integrated with a small dataset from Portugal, consisting of three subjects, of which two with ASD and the other was the therapist. Each subject executed IOGIOCO protocol, including level 3, several times. The segmentation process was similar to the one described in the previous section. The higher number of samples for each gesture allowed to consider the entire gesture set (19 gestures).

3.3 Data Processing

When the whole dataset was ready, each file was processed. Keypoints files contain in each row a timestamp t (seconds from epoch) followed by 25 skeleton joints (x_k, y_k, z_k coordinates for each k skeleton joint) describing the human skeleton of every frame as shown in Equation 3.1.
Frame F_t:

$$t \;\; x_0 \;\; y_0 \;\; z_0 \;\; ... \;\; x_{24} \;\; y_{24} \;\; z_{24} \tag{3.1}$$

3.3.1 Translation-Invariance and User-Invariance

All coordinates were adjusted to a new reference system and normalized frame by frame. Each coordinate was first referenced to a particular keypoint to make it translation-invariant:

$$\begin{aligned} x_{new} &= x - x_r, \\ y_{new} &= y - y_r, \\ z_{new} &= z - z_r, \end{aligned} \tag{3.2}$$

where $[x_{new}, y_{new}, z_{new}]$ are the coordinates in the new reference system and $[x_r, y_r, z_r]$ are the ones of the reference keypoint. Then, in order to make each dataset user-invariant (i.e. scale-invariant), each coordinate was normalized by h. The value h was computed in each frame as the Euclidean distance between two coordinates $\mathbf{k_1}$ and $\mathbf{k_2}$ representing the subject anatomical characteristics:

$$h(\mathbf{k_1}, \mathbf{k_2}) = \sqrt{(x_{k_1} - x_{k_2})^2 + (y_{k_1} - y_{k_2})^2 + (z_{k_1} - z_{k_2})^2} \tag{3.3}$$

with $[x_{k_1}, y_{k_1}, z_{k_1}]$ and $[x_{k_2}, y_{k_2}, z_{k_2}]$ as the two keypoints coordinates. In this way, each file was independent of the person doing the gesture. The value h could not reflect the real body size of the subject because of his undefined position in front of the Kinect camera, which has low spatial resolution on depth data [12]. For these reasons, it was not convenient to set h at a fixed value at the beginning of the therapy session and a frame by frame normalization was chosen. Inspired by [57], Shoulder Center keypoint and the total height of each subject were first selected as reference point and h, respectively. The total height was computed as the Euclidean distance between head keypoint and the middle point between left and right foot. Then, considering this project's characteristics, different combinations were tested (Table 3.2).

Table 3.2: Reference keypoints, normalization's lengths and respective $\mathbf{k_1}$ *and* $\mathbf{k_2}$ *keypoint's coordinates. Note that the height was computed as the Euclidean distance between head keypoint 3 and the point 25* computed as the middle point between left and right foot.*

Reference Keypoint	**h**	**$\mathbf{k_1}$ $\mathbf{k_2}$ keypoints**
Shoulder Center	Height	3-25*
Shoulder Center	Head-Trunk	3-0
Hip Center	Head-Trunk	3-0
Hip center	Trunk	20-0

3.3.2 Filtering

After, a median filter was implemented over a 5-frames window to reduce the impulsive noise: each coordinate was replaced with the median value of the coordinate in the previous window:

$$\begin{aligned} x_o &= Median(x_i), \\ y_o &= Median(y_i), \\ z_o &= Median(z_i), \end{aligned} \tag{3.4}$$

where $[x_o, y_o, z_o]$ are the output signals and $[x_i, y_i, z_i]$ are the input ones.

3.4 Pose Features

Human body reconstruction is needed in order to highlight the information characterizing a particular gesture in a single sample and to discard less important details. When using ANN and the only available motion features are skeletal data, an intermediate image representation of skeletal sequences can help in data processing and in looking at samples the net has to learn from. Each sequence representing an action is formed by F_1 to F_N frames. Following the work of Pham et al. [9, 33], every frame F_t of a sequence was transformed in a 3D array to be stacked in a matrix together with the others. The matrix containing the 3D coordinates (x_k, y_k, z_k) of all frames, with k ranging from 0 to K body joints, was rearranged according to human physical structure. A single action sequence was then normalized by a normalization function $\mathbf{N}(\cdot)$ and denotes a *pose feature*. Finally, a single *RGB pose feature* representing the motion was obtained through a transformation function $\mathbf{G}(\cdot)$. The entire process is shown in Figure 3.7.

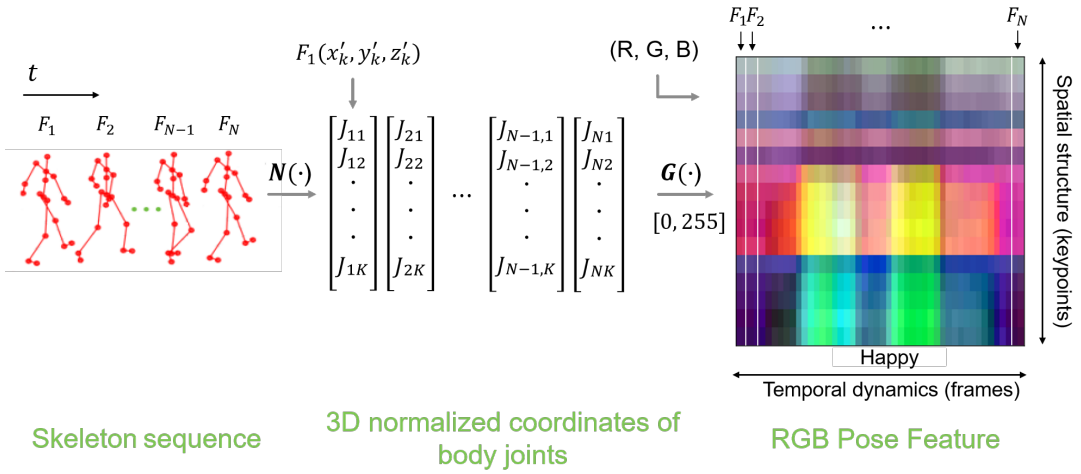

Figure 3.7: Illustration of the data transformation process. Every frame F_t of a sequence has been transformed in a 3D array to be stacked in a matrix together with the others. N denotes the number of frames in each sequence and K denotes the number of keypoints in each frame. Each skeleton sequence is normalized by a normalization function $\mathbf{N}(\cdot)$ to obtain a pose feature. Then each pose feature is transformed in a single RGB pose feature representing the motion through a transformation function $\mathbf{G}(\cdot)$ to get a skeleton-based representation. On the horizontal dimension temporal dynamics is shown, while the spatial structure (keypoints) is depicted on the vertical one.

Starting from this skeleton-based representation, further data processing was implemented. The following subsections describe the entire process.

3.4.1 Rearrangement of Body Keypoints

Skeleton joints were ordered in each frame F_t according to human body physical structure to have an effective representation of each gesture and to keep the local motion characteristics. In this way, more discriminating features easily distinguishable by the learning model were generated. Inspired by [35] and [33], each skeleton frame was rearranged into five parts: two arms, two legs and one trunk. Since the lower body was not crucial in this therapy protocol's gesture set, only the upper body segments were preserved. In particular, 16 out of 25 keypoints were selected to represent the gestures and their action kinematics. They were grouped in body sets: head and trunk, right arm and left arm. As shown in Table 3.3, each of them was defined by physically ordered keypoints.

Table 3.3: Body sets whose keypoints are detailed in the Joint's Dictionary of Table 3.1.

Body Sets	**Related Keypoints**
Head and Trunk	3, 2, 20, 4, 8, 1
Right Arm	9, 10, 11, 23, 24
Left Arm	5, 6, 7, 21, 22

Moreover, body sets were organized from top to bottom in different combinations (for instance, head and trunk first, right arm and left arm then) in order to find the best keypoints' arrangement for the algorithm to learn. To evaluate how the network reacts to these combinations, different body sets orders were used to train, validate and test the model. Experiments were done on the Healthy Dataset.

3.4.2 From Body Keypoints to Pose Features

In order to describe the temporal dynamics, every action sequence related to a gesture, which denotes the biomechanics of skeletons, was represented into a 3D matrix by stacking together each time frame. This 3D matrix is the reconstruction of the human pose during the movement. To obtain a *pose feature*, all the 3D coordinates (x_k, y_k, z_k) of each frame F_t in a sequence were scaled through a normalization function $\mathbf{N}(\cdot)$:

$$\begin{aligned}(x'_k, y'_k, z'_k) &= \mathbf{N}(x_k, y_k, z_k) \\ x'_k &= \frac{(x_k - x_{min})}{(x_{max} - x_{min})}, \\ y'_k &= \frac{(y_k - y_{min})}{(y_{max} - y_{min})}, \\ z'_k &= \frac{(z_k - z_{min})}{(z_{max} - z_{min})},\end{aligned} \tag{3.5}$$

where (x'_k, y'_k, z'_k) are the normalized coordinates of k-th keypoint, $\mathbf{c_{max}} = (x_{max}, y_{max}, z_{max})$ and $\mathbf{c_{min}} = (x_{min}, y_{min}, z_{min})$ are the scaling coordinates as described in the following Subsection 3.4.3 Gestures Normalizations. As shown in Figure 3.8, a pose feature is composed

by keypoints' position in space at each time instant on the vertical axis and by the series of frames characterizing the action's range on the horizontal axis.

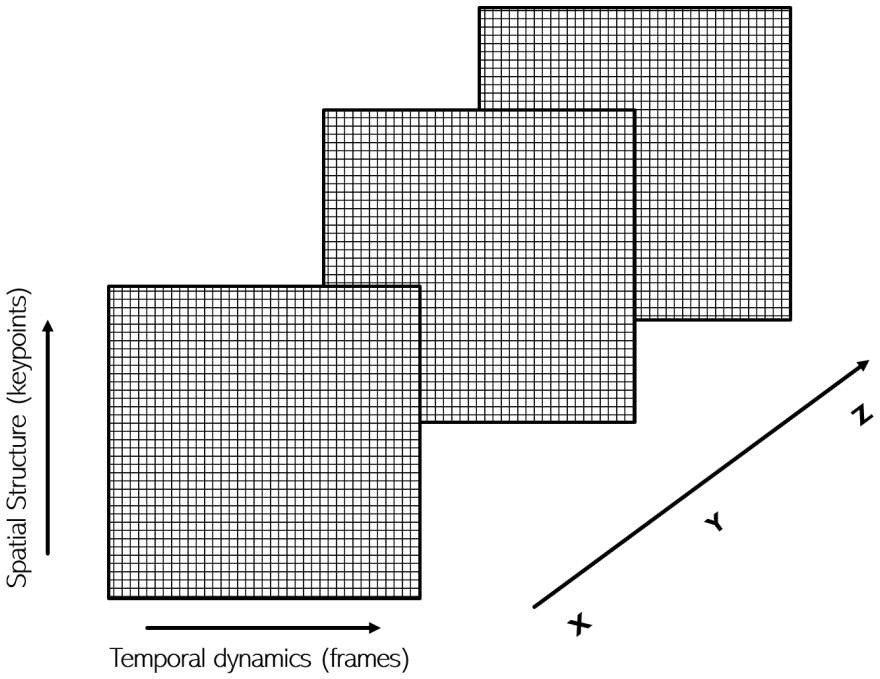

Figure 3.8: Representation of the 3D pose feature.

3.4.3 Gestures Normalizations

All coordinates in each sample of each dataset were normalized by the function $\mathbf{N}(\cdot)$, using maximum and minimum coordinates to set the spatial range in which a gesture was performed. Different gestures normalizations have been tested:

- Gesture-dependent vs gesture-independent normalization;
- Per keypoint vs per body control volume normalization.

In *gesture-dependent* normalizations, each coordinate could be scaled differently depending on the type of action executed: maximum and minimum coordinates characterizing each gesture were selected over the entire dataset. In *gesture-independent* coordinates normalizations, maximum and minimum coordinates of each channel (x, y, z) were detected during a movement, whatever gesture executed and used as scaling values. The first normalization makes samples more user-invariant since it defines an action-specific displacement range equal for all subjects by computing the scaling coordinates with respect to the entire dataset, thus being dataset-specific. Instead, the second one creates a different bounding box for each movement executed, thus allowing a dataset-independent approach.

Per keypoint normalization implies that movements are scaled in terms of maximum and minimum displacement range for each body joint while *per body control volume* normalization means that a whole-body control volume was defined and maximum and minimum coordinates are selected independently of body joints.

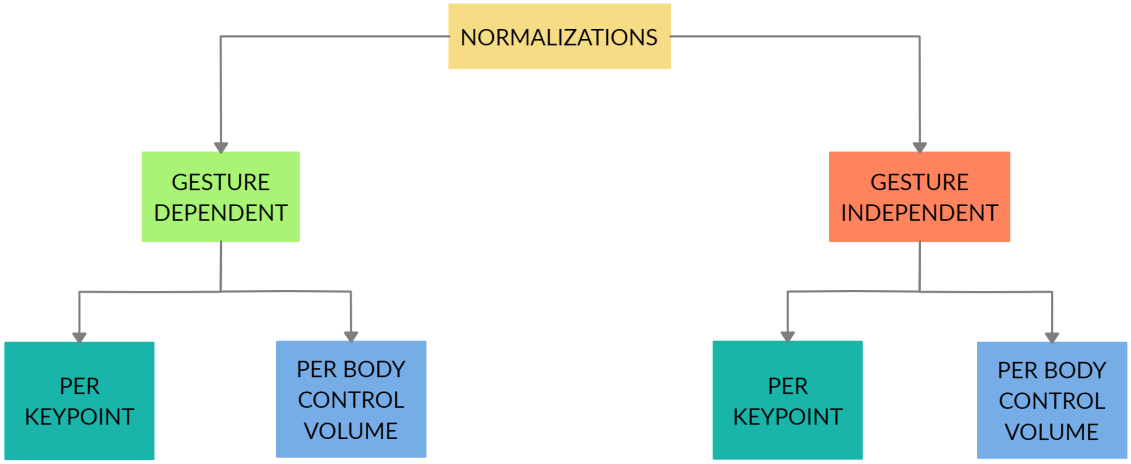

Figure 3.9: Normalizations' Map.

Gesture-dependent normalization could not be replicated during algorithm test phase since gestures were not recognized yet. Therefore, in such cases, test set normalizations were carried out without taking into account the type of gesture executed. These trials were reasonable because, even though *gesture-independent* normalization is user-variant, it normalizes the action itself, which actually is gesture-specific. Normalizations were experimented on the Healthy Dataset and summarized in Table 3.4.

Table 3.4: Gesture Normalizations experiments.

Train and Validation	**Test**	
gesture-independent	gesture-independent	per body control volume per keypoint
gesture-dependent	gesture-independent	per body control volume per keypoint

Gesture-dependent Normalization

- *Per keypoint*: maximum and minimum coordinates were computed for each keypoint and for each type of gesture, over the entire dataset:

$$\begin{aligned} \mathbf{c_{max}}[g][k] &= (x_{max}[g][k], y_{max}[g][k], z_{max}[g][k]), \\ \mathbf{c_{min}}[g][k] &= (x_{min}[g][k], y_{min}[g][k], z_{min}[g][k]), \end{aligned} \tag{3.6}$$

  where g is the distinct gesture and k is the k-th keypoint.

  Each sample $\mathbf{p}$ was then scaled by $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ according to Equation 3.7.

$$\mathbf{coord}[g][k] = \frac{\mathbf{coord}[g][k] - \mathbf{c_{min}}[g][k]}{\mathbf{c_{max}}[g][k] - \mathbf{c_{min}}[g][k]}. \tag{3.7}$$

- *Per body control volume*: maximum and minimum coordinates were computed for

each type of gesture considering all body joints, over the entire dataset:

$$\begin{aligned} \mathbf{c_{max}}[g] &= (x_{max}[g], y_{max}[g], z_{max}[g]), \\ \mathbf{c_{min}}[g] &= (x_{min}[g], y_{min}[g], z_{min}[g]), \end{aligned} \tag{3.8}$$

where g is the distinct gesture.

Each sample $\mathbf{p}$ was then scaled by $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ according to Equation 3.9.

$$\mathbf{coord}[g][k] = \frac{\mathbf{coord}[g][k] - \mathbf{c_{min}}[g]}{\mathbf{c_{max}}[g] - \mathbf{c_{min}}[g]} \tag{3.9}$$

Gesture-independent Normalization

- *Per keypoint*: maximum and minimum coordinates were computed on each sample $\mathbf{p}$ for each keypoint without taking into account the type of gesture:

$$\begin{aligned} \mathbf{c_{max}}[k] &= (x_{max}[k], y_{max}[k], z_{max}[k]), \\ \mathbf{c_{min}}[k] &= (x_{min}[k], y_{min}[k], z_{min}[k]), \end{aligned} \tag{3.10}$$

  where k is the k-th keypoint.

  Each sample $\mathbf{p}$ was then scaled by $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ according to Equation 3.11.

$$\mathbf{coord}[k] = \frac{\mathbf{coord}[k] - \mathbf{c_{min}}[k]}{\mathbf{c_{max}}[k] - \mathbf{c_{min}}[k]} \tag{3.11}$$

- *Per body control volume*: maximum and minimum coordinates were computed on each sample $\mathbf{p}$ considering all body joints without taking into account the type of gesture:

$$\begin{aligned} \mathbf{c_{max}} &= (x_{max}, y_{max}, z_{max}), \\ \mathbf{c_{min}} &= (x_{min}, y_{min}, z_{min}). \end{aligned} \tag{3.12}$$

  Each sample $\mathbf{p}$ was then scaled by $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ according to Equation 3.13:

$$\mathbf{coord}[k] = \frac{\mathbf{coord}[k] - \mathbf{c_{min}}}{\mathbf{c_{max}} - \mathbf{c_{min}}} \tag{3.13}$$

In order to analyze the difference between *gesture-dependent* and *gesture-independent* normalizations, some parameters were computed:
Gesture-dependent
For training set, the Euclidean distance d between the maximum and minimum coordinates $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ considering the total movements' kinematics for a particular gesture class $\mathbf{g}$ over the entire dataset was computed according to Equation 3.14.

$$d(\mathbf{c_{max}}[\mathbf{g}], \mathbf{c_{min}}[\mathbf{g}]) = \sqrt{(x_{max}[\mathbf{g}] - x_{min}[\mathbf{g}])^2 + (y_{max}[\mathbf{g}] - y_{min}[\mathbf{g}])^2 + (z_{max}[\mathbf{g}] - z_{min}[\mathbf{g}])^2} \tag{3.14}$$

Gesture-independent:
For test set, the Euclidean distance d between the maximum and minimum coordinates $\mathbf{c_{max}}$ and $\mathbf{c_{min}}$ of a particular sample $\mathbf{p}$, whatever gesture class, was computed according to Equation 3.15.

$$d(\mathbf{c_{max}}[\mathbf{p}], \mathbf{c_{min}}[\mathbf{p}]) = \sqrt{(x_{max}[\mathbf{p}] - x_{min}[\mathbf{p}])^2 + (y_{max}[\mathbf{p}] - y_{min}[\mathbf{p}])^2 + (z_{max}[\mathbf{p}] - z_{min}[\mathbf{p}])^2} \tag{3.15}$$

3.4.4 RGB Pose Features

The normalized pose features were then converted into RGB matrices to get a straight forward visual representation. All the 3D coordinates (x_k, y_k, z_k) of each frame F_t in a sequence were transformed into a new color space through a transformation function $\mathbf{G}(\cdot)$:

$$\begin{aligned}(x''_k, y''_k, z''_k) &= \mathbf{G}(x'_k, y'_k, z'_k),\\ x''_k &= 255 \times x'_k,\\ y''_k &= 255 \times y'_k,\\ z''_k &= 255 \times z'_k,\end{aligned} \tag{3.16}$$

where (x''_k, y''_k, z''_k) are the coordinates of k-th keypoint in the new color space and (x'_k, y'_k, z'_k) are the normalized coordinates as described in Subsection 3.4.2 From Body Keypoints to Pose Features. Hence, coordinates were coded into RGB color space, which means scaled between 0 and 255. The schema of the RGB pose feature is shown in Figure 3.10. In this

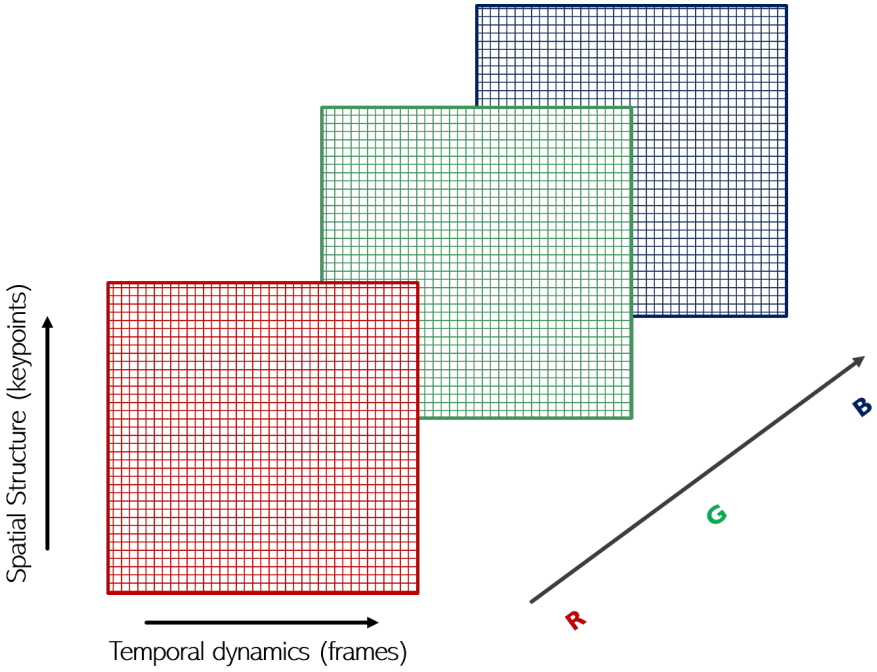

Figure 3.10: Representation of the RGB pose feature.

way, kinematics of each action was kept and highlighted by a new representation. During a movement, a displacement in the x direction is depicted by a variation of red amount, while a shift in y or z direction corresponds to a change in green or blue, respectively. Therefore, each skeleton joint in a certain instant was represented by a single pixel in a 2D image. This image representation made pose features processing possible as described in Subsections 3.4.5 and 3.4.6.

3.4.5 Temporal Interpolation and Reshape

The input images should have the same resolution/size (height and width) in order to be processed by a neural network. Thus, different ways of resizing were experimented. For

simplicity, to use symmetric pooling and convolutions during net training, a squared inputs' format was exploited.

At the beginning, inspired by Pham et al. [9], a resize of 32×32 pixels on each pose feature was experimented. To this purpose, the Nearest Neighbour resampling filter was applied to the whole image. The nearest pixel to the interpolated point is picked from all adjacent pixels in the source image; Figure 3.11 (a) illustrates how this interpolation works in the upsampling case. Secondly, the resampling process was changed in order to maintain keypoint's information distinct one from the other and to involve the temporal dynamics only. In this way, the nearest pixel is picked only from the previous one and the next one in the source image (Figure 3.11 (b)). Thus, the Nearest Neighbour resampling filter was applied row by row to obtain a *Temporal Interpolation*.

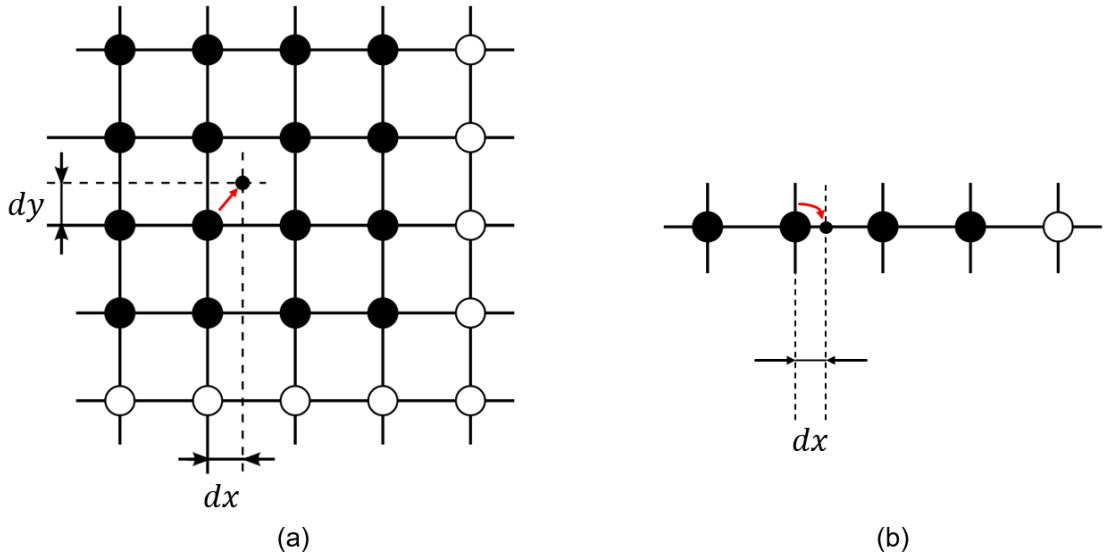

Figure 3.11: (a) Whole image interpolation. Big circles represent existing image pixels. The small dot stands for the new pixel that have to be created in the scaled image and dx and dy are the offsets defining its interpolated position with respect to the nearest pixel; (b) Row by row interpolation. The interpolated point gets the nearest pixel value which, in this case, will be either the previous or the next one. In both the figures the arrow shows the assignment of the closest pixel to the re-sampling.

As already mentioned in Subsection 3.4.1 Rearrangement of Body Keypoints, the 16 selected keypoints were represented in the pose feature on the vertical dimension. Since a 16×16 resolution would have been too low, a better one was obtained by stacking together each keypoints' row three times ($aaa - bbb - ccc$ configuration, Figure 3.12) or triple the action pose feature ($abc - abc - abc$ configuration, Figure 3.13).

3.4.6 Enhanced Action Images

In order to highlight each RGB pose feature, contrast of images was enhanced exploiting Contrast Limited Adaptive Histogram Equalization (CLAHE), an upgraded version of the one used in [34]. Given that CLAHE operates on small regions of the image called tiles, *local* contrast is amplified. Moreover, if any histogram bin is above a specified contrast threshold called Clip Limit, those pixels are redistributed uniformly to other bins before

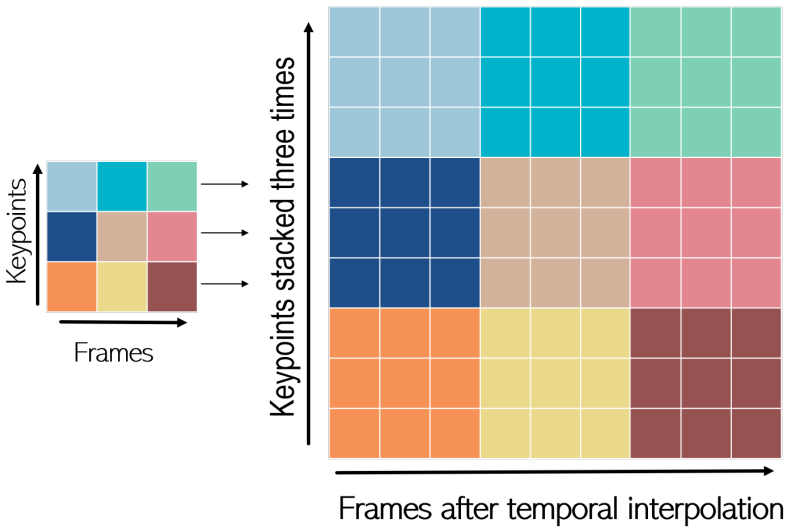

Figure 3.12: Illustration of a pose feature after temporal interpolation with 48×48 resolution and keypoints' row stacked together three times, $aaa - bbb - ccc$ configuration.

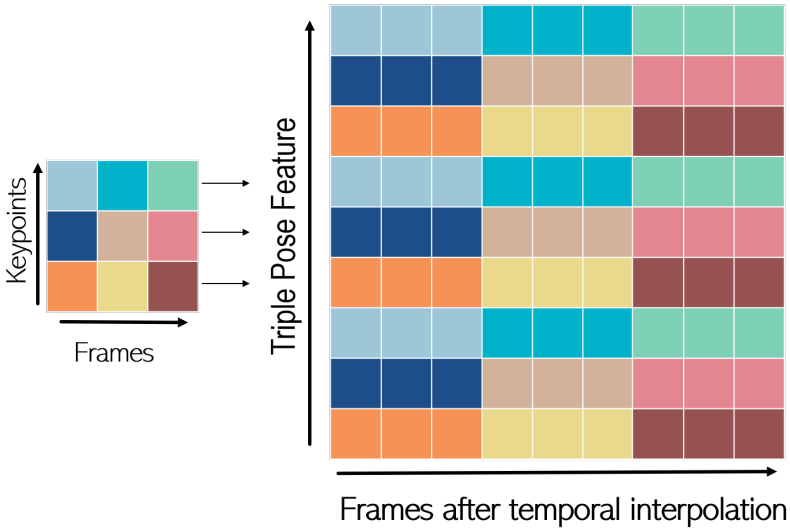

Figure 3.13: Illustration of a pose feature after temporal interpolation with 48×48 resolution and tripling, $abc - abc - abc$ configuration.

applying histogram equalization (Figure 3.14). A Clip Limit of 0.01 and a tile of 1×48

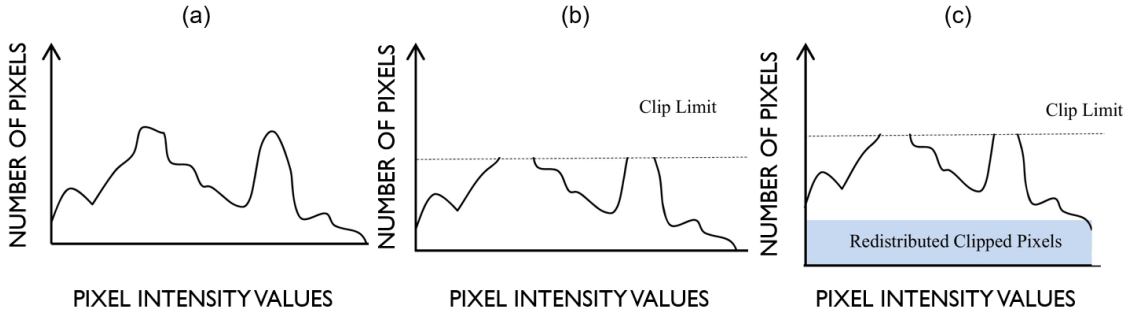

Figure 3.14: Clipped Limited Histogram Equalization method. (a) Histogram of the original input image. (b) Clipping the histogram based on predefined Clip Limit. (c) Modified histogram after redistribution of the Clipped portion.

were exploited on every row (i.e. every keypoint's dynamics) to keep the spatial structure of the pose feature.

3.4.7 Net Inputs Preparation

As already mentioned in 3.4.4 RGB Pose Features, RGB matrices were a image representation of actions' kinematics. Furthermore, pixel values are unsigned integers in the range between 0 and 255. Hence, feeding this raw format pixels directly to neural network models can result in a slower training of the model. Instead, there can be benefit in preparing the image pixel values prior to training, scaling pixel values to the range 0-1 standardizing the values [58]. Thus, the net's input were re-scaled to get *pose features* again, according to the following equation:

$$\begin{aligned} x'_k &= \frac{x''_k}{255}, \\ y'_k &= \frac{y''_k}{255}, \\ z'_k &= \frac{z''_k}{255}. \end{aligned} \tag{3.17}$$

3.4.8 Data Mirroring

This project's datasets were composed by actions performed with the right arm. In order to make the Expanded Dataset more generalized, the original movements were mirrored. Thus, the x coordinates of each frame were mirrored with respect to the *Reference Point* used. In this way, for one-limb gestures, the algorithm could learn to recognize them independently of the dominant hand. For symmetric two-limbs gestures this represented a data augmentation process.

3.4.9 Datasets Split

In order to train the recognition algorithm, datasets were split into training set, validation set and test set each.

Subsampled Healthy Dataset Split

Since all gestures were performed 14 times by every subject of this dataset (balanced dataset), the algorithm was trained, validated and tested on all subjects. For each gesture performed by a specific subject, 10 files out of 14 were used for training, 2 out of 14 for validation and 2 out of 14 for testing (Table 3.5).

Table 3.5: Subsampled Healthy Dataset Split.

	Split	**Samples (%)**
Train	10 files × 18 people × 5 gestures	900 (∼ 71.4%)
Validation	2 files × 18 people × 5 gestures	180 (∼ 14.3%)
Test	2 files × 18 people× 5 gestures	180 (∼ 14.3%)
Tot		1260

Healthy Dataset Split

This dataset was unbalanced because the 18 subjects executed a different number of repetition per gestures and not all actions in the gesture set were executed. Therefore, two people among the subjects with all types of gestures performed were left out for testing. The samples of the rest of the subjects were merged independently of the person in order to create balanced training and validation sets.

*Table 3.6: Healthy Dataset Split. *Note that, for testing, two people unseen during training with an unbalanced number of files per gesture were selected.*

	Split	**Samples (%)**
Train	20 files × 14 gestures	280 (∼ 75%)
Validation	3 files × 14 gestures	42 (∼ 11%)
Test	2 people's files*	45 (∼ 14%)
Tot		367

Expanded Healthy Dataset Split

The expanded dataset was composed by all the 19 gestures with 2 ASD out of 22 subjects, but still unbalanced. As in the previous dataset split, two people among N subjects with all type of gestures performed were left out for testing. A Leave-P-Out subject cross-validation method was exploited for this more comprehensive 19-gestures dataset. In this way, P out of N sybjects in the dataset were used for testing and P-N for training and validating the model (P=2 subjects and N=11 subjects). Different net's hyperparameters were tested to achieve the best recognition results possible. The samples of the rest of the subjects were merged independently of the person in order to create balanced training and validation sets. Table 3.7 shows train, validation and test samples split.

*Table 3.7: Expanded Healthy Dataset Split. *Note that, for testing, two people unseen during training with an unbalanced number of files per gesture was used. Moreover, two ASD subjects where part of the dataset.*

	Split	**Samples (%)**
Train	18 files × 19 gestures	342 (∼70%)
Validation	5 files × 19 gestures	95 (∼20%)
Test	2 people's files*	50 (∼10%)
Tot		487
Tot after data mirroring		487 × 2

3.5 Classification

In [33, 9, 34], Pham et al. proposed deep residual learning with residual blocks for recognizing human action from skeleton sequences. Inspired by this approach we implemented a deep learning framework based on ResNet. ResNet allowed the design of a deeper neural network able to resolve the vanishing gradient problem, without degradation in performance.

3.5.1 Neural Network design

Based on what has been said about ResNets in Subsection 2.3.4 and following the paper of Pham et al. [9], a similar architecture was designed (Figure 3.15). The network starts with a Convolutional (Conv) layer with $K \times K$ filters,a ReLU and a Batch Normalization (BN) layer, followed by 3 ResNet stages, an Average Pooling layer and a Dense layer for the final classification (Figure 3.15(a)). The first stage consists of n Identity Residual Blocks. The second and the third stages consist of one Convolutional Residual Block and $n - 1$ Identity Residual Blocks. For the reasons given in Subsection 2.3.4, after each stage, the number of filters is doubled. An Identity Residual Block is formed by Conv-ReLU-BN-Conv-ReLU-BN layers with the shortcut connection added to the output before another ReLU layer (Figure 3.15(b)). A Convolution Residual Block is characterized by a first convolutional layer with stride of 2 followed by ReLU-BN-Conv-ReLU-BN layers. The shortcut connection added to the output before another ReLU layer is characterized by a 1×1 Convolutional layer (Figure 3.15(c)).

3.5.2 Hyperparameters Tuning

To achieve the best results, net's hyperparameters had to be set. To this purpose, Ax, an experimentation platform able to optimize any kind of experiment, was exploited. Once an experiment was launched, multiple trials were performed. Each trial evaluated the possible combinations of hyperparameter values through a 'score' function. The mean of the validation accuracies of the last ten epochs was chosen as score. Since Ax tracks the history of parameters and scores, the best set of hyperparameters, corresponding to the highest output score, was retrieved. The different options of hyperparameters chosen for Subsampled Healthy Dataset are shown in Table 3.8, while the ones for both Healthy Dataset and Expanded Dataset are shown in Table 3.9.

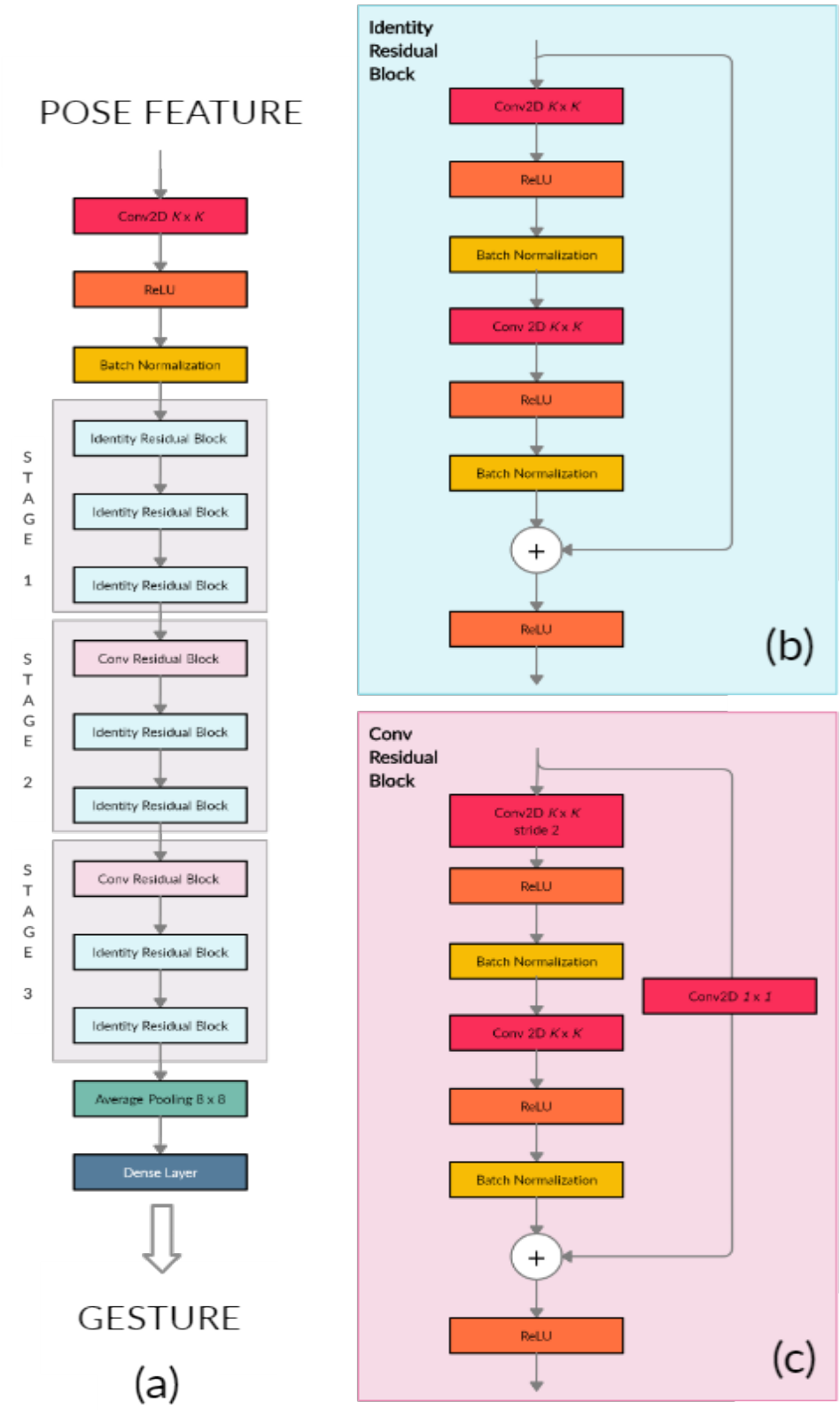

Figure 3.15: ResNet design.

Table 3.8: Subsampled Healthy Dataset hyperparameters.

Hyperparameters	**Options**
Number of Res Blocks	3, 5, 6, 9
Batch size	8, 16, 32, 64, 128, 256
Activation	Sigmoid, ReLU
Optimizer	Adam, SGD, RMS

Table 3.9: Healthy Dataset and Expanded Dataset hyperparameters.

Hyperparameters	**Options**
Number of Res Blocks	3, 5, 6, 9
Batch size	8, 16, 32, 64, 128, 256
Activation	Sigmoid, ReLU
Optimizer	Adam, SGD, RMS
Number of Filters	4, 8, 16, 32
Kernel Size of filters	3, 5, 9, 11

3.6 Online Recognition

Once the algorithm was established offline, an online implementation was designed to be integrated in Level 3 of the therapy protocol. As mentioned in Subsection 3.1.2, the subject performs the gesture after NAO points at her/him and the recognition task must occur within seconds. This implementation was first tested using only Kinect camera, and then integrated with the robot:

- Kinect-only configuration: the model was set and tested on the continuous data stream captured by the camera;
- Kinect-NAO configuration: the model was set and tested with the robot.

In fact, when dealing with physical robots timing have to be taken into account and Kinect's behaviour changes. For this reason, Kinect camera's FPS was evaluated in both configurations.

3.6.1 Kinect-only Settings

In order to exploit the recognition algorithm in a real-time classification, a sliding window was used. Pose features were computed and analyzed by the classifier on a certain window, characterized by two configuration parameters: *size* and *step*. A fixed *size* of the window in terms of number of frames was used on the continuous data stream captured by the camera. Different configurations were experimented to avoid lag between the performance of the action and the classification's output as much as possible. In the end, the average number of frames for each gesture class was calculated. So, the window *size* was set to the mean value over all gesture classes. To detect the presence of a gesture among *no gestures*, the highest conditional probability output by the Softmax layer of the classifier

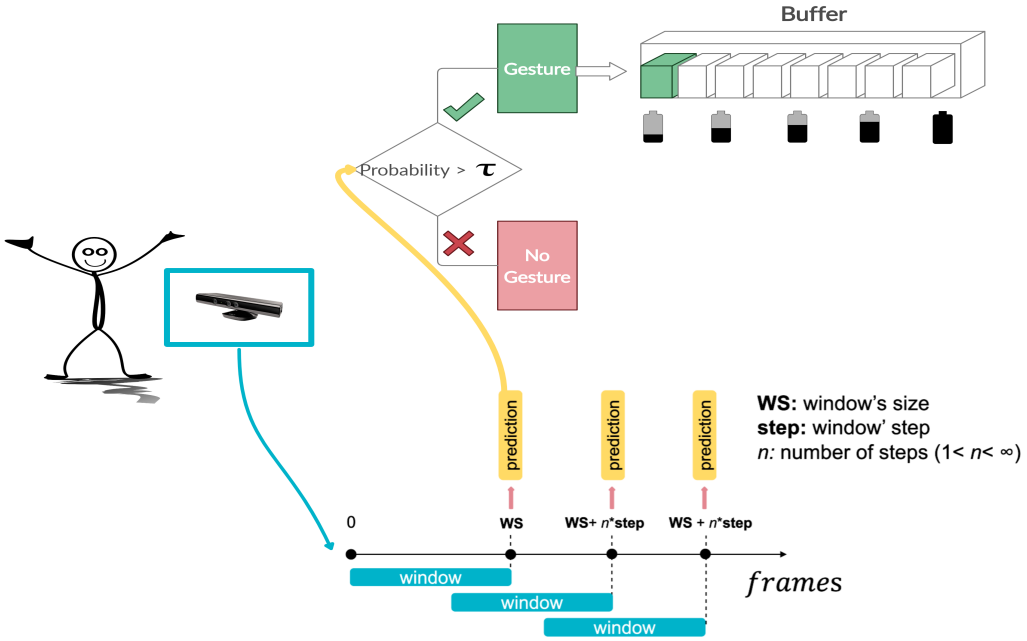

Figure 3.16: Gesture vs no Gesture. Kinect camera captures frames; when the window is filled with the proper number of frames (WS, window's size), a prediction is output. If the highest conditional probability exceeds the threshold, a gesture is detected and the probabilities' prediction vector is saved in the buffer. The window slides of a fixed step and the process re-start.

was compared to a threshold $\tau \in [0,1]$:

$$state = \begin{cases} gesture, & \text{if } probability > \tau \\ no - gesture, & \text{if } probability < \tau \end{cases} \tag{3.18}$$

Threshold value τ was chosen as the minimum softmax prediction probability among true positive's predictions (true label equal to predicted label). When the detection threshold was exceeded, the probabilities' prediction vector was saved in a *buffer*. Then, the window slided of a fixed *step* before predicting again (Figure 3.16). Once the *buffer* was filled with N prediction vectors, the algorithm identified the gesture performed with one of the following two possible methods:

- By averaging *buffer*'s prediction vectors' probabilities;
- By checking whether all *buffer*'s predictions were equal.

Moreover, in Kinect-NAO configuration, a positive or negative sound feedback was implemented on the robot to be given as an output depending on the performance assessment. Once the whole algorithm was established, new acquisitions were performed to test the effectiveness of the new method.

3.6.2 Kinect-NAO Settings

The next step was characterized by the integration of the recognition algorithm with NAO robot. In this online version with NAO, the window of fixed size sliding on the continuous data stream from Kinect and the buffer mentioned in the previews Subsection 3.6.1 were

preserved. Since Kinect's performance changed with NAO connection, new parameters were tested. Moreover, according to the protocol, a sound feedback was implemented on the robot, to be given as an output. To ensure the feedback was triggered when the gesture was completed, the algorithm started predicting only after a fixed amount of time. A flag was also inserted in order to prevent the same action movement from being classified multiple times and to make NAO speak only once. In this way, the algorithm had a *cool down* time interval. Once the algorithm was established, new acquisitions were performed to test the effectiveness of the new method.

3.7 Acquisitions

To evaluate the effectiveness of the new algorithm, new acquisitions were carried out both at NearLab in Politecnico di Milano and at CARElab (Computer Assisted Rehabilitation) in Fondazione Don Gnocchi.

3.7.1 @Politecnico Acquisitions

Before starting with Kinect-only and Kinect-NAO configuration's acquisitions, to analyze prediction's vector probabilities' trend and to estimate the effectiveness of online settings' implementation, *tall*, *hello* and *little* gestures where performed by an healthy subject in Kinect-only configuration. Kinect and robot gesture recognition performances were then analyzed through acquisitions on two healthy subjects in Kinect-only and Kinect-NAO configurations. Each subject correclty performed all gestures 2 times mimicking the therapy protocol and complying with NAO timings in the case of Kinect-NAO configuration. Accuracy, Precision, Recall, F1-score have been computed for all acquisitions according to Equations 3.19, where TP = True Positives, TN = True Negatives, FP = False Positives and FN = False Negatives.

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TN + FP + FN + TP} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F - score &= \frac{2 \times Recall \times Precision}{Recall + Precision} \end{aligned} \tag{3.19}$$

3.7.2 @CARElab Acquisitions

In weekly sessions of about 10−20 minutes, 6 ASD children aged between 4 and 6 were part of IOGIOCO therapy-protocol. Each session was developed in a room as empty as possible to avoid distractions. Kinect camera was placed above a television, while NAO Robot was sat in the room centre, so that it could be seen as soon as children entered in the room (Figure 3.17). All data detected by Kinect were saved toward data analysis: videos, keypoints' files and gesture algorithm's predictions. Accuracy, Precision, Recall and F-score were computed for all acquisitions. Since in clinical acquisitions gestures were performed a different number of times, the class distribution was uneven. For this reason, F1-score was a better measure of the incorrectly classified cases than the accuracy metric.

Figure 3.17: CARElab therapy room's set up.

Chapter 4

Results

In this chapter results for the validation of the recognition algorithm are presented. Acquisitions done with healthy adults and with ASD children are analyzed at the end of the chapter.

4.1 Data Processing

4.1.1 Translation-Invariance and User-Invariance

Right after data acquisition, skeletal data was referenced to a particular keypoint and normalized by h, computed frame by frame, in order to make the algorithm user-invariant. Table 4.1 summarize the experiments carried out. The reference keypoint was changed

Table 4.1: Reference keypoints and normalization's lengths.

Reference Keypoint	**h**
Shoulder Center	Height
Shoulder Center	Head-Trunk
Hip Center	Head-Trunk
Hip center	Trunk

from Shoulder Center to Hip Center for stability issues: in fact, given that the protocol's gesture set involves the upper body mainly, hip joint's position is characterized by less movements and variations in position.

The first normalization experimented was the one suggested by [57], in which h was the total height of the subject. However, in this project the type of normalization and the tools exploited for data acquisition must be taken into account. So, the Standard Deviation (SD) of different values of h along frames over the entire dataset was analyzed. In Figure 4.1, the mean of the SD of normalization segments of all samples is shown for each gesture through column charts. From the two charts it's possible to notice that the most unstable segment during the performances of almost all gestures was the height one. This was due to Kinect system larger noise behaviour in feet and ankles. Instead, head and trunk and shoulder-shoulder segments had a lower Standard Deviation, since their computation does not involve the bottom part of the body. However, these normalization segments were not

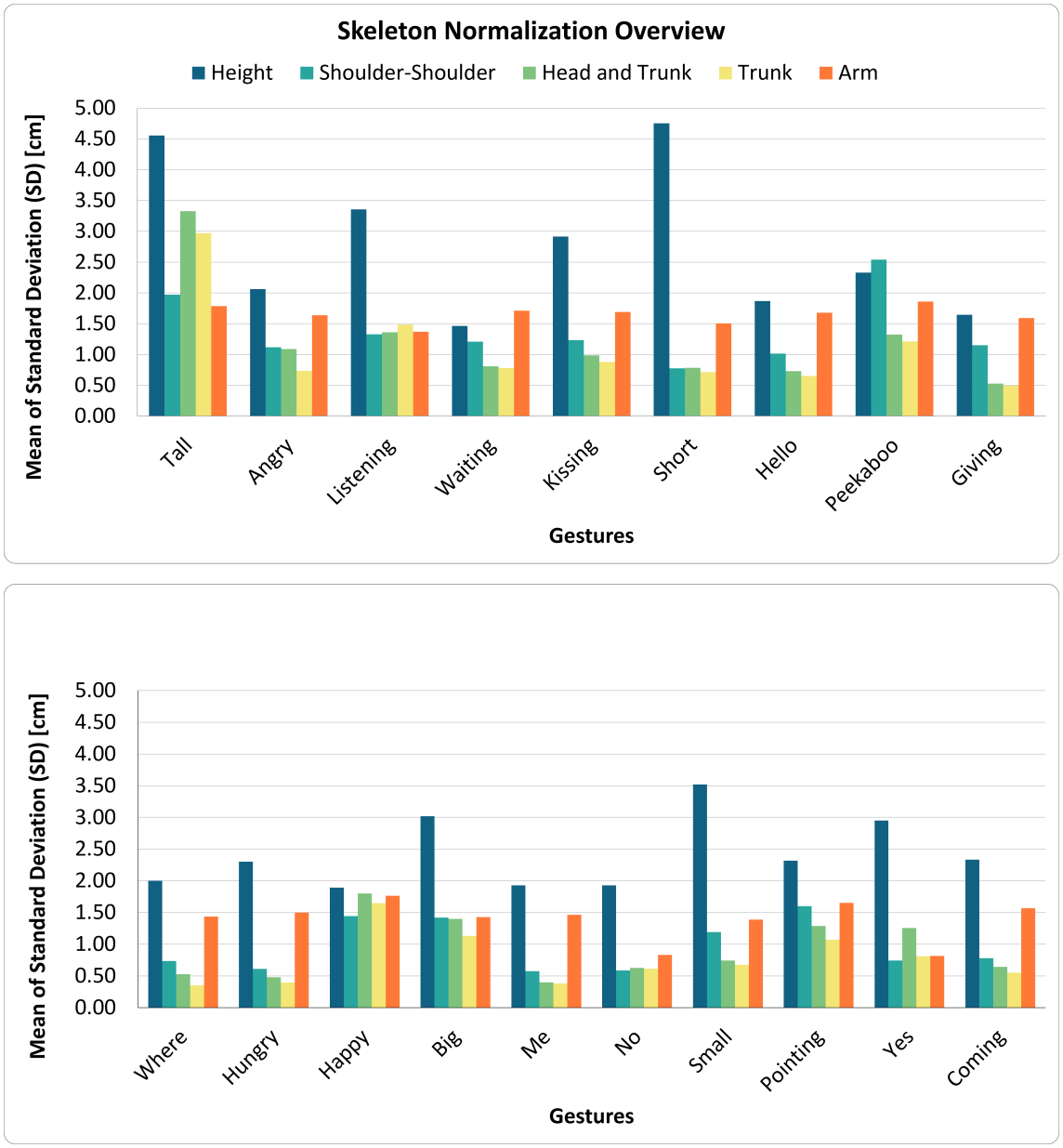

Figure 4.1: The mean of the Standard Deviation of different normalization's segments for each class in the Expanded Dataset is shown through two column charts.

the most stable due to head and shoulders movements while performing gestures. The arm length's mean Standard Deviation was quite the same for all actions, but still high. As a result, the trunk size turned out to be the most stable length during the performance of an action for almost all gestures, since is the least action-involved segment.

4.2 Pose Features

4.2.1 Rearrangement of body keypoints

Different body sets orders were tested to find the best keypoints' arrangement for the algorithm to learn, as shown in Table 4.2.

Table 4.2: Upper body sets orders tested.

Shortening	**Body Sets orders**
TRL	Head and trunk, Right limb, Left limb
TLR	Head and trunk, Left limb, Right limb
RTL	Right limb, Head and trunk, Left limb
RLT	Right limb, Left limb, Head and trunk
LTR	Left limb, Head and trunk, Right limb
LRT	Left limb, Right limb, Head and trunk

The respective RGB pose features describing the kinematics of *pointing* gesture are shown in Figure 4.2. For each RGB pose feature, keypoints are on the vertical dimension, while the temporal dynamic is represented on the horizontal dimension. As can be noticed, each pose feature is characterized by a color changing from pink to yellow in correspondence to keypoints of the right limb. In fact, *pointing* gesture was executed by this subject (part of Expanded Dataset) with the right arm. Joints' movement with respect to the *Reference point* is depicted by a color changing. The amount of red, green or blue changed depending on the direction of the action (red = x direction, green = y direction, blue = z direction). This variation is depicted in a different part of the image depending on body set orders organization.

In order to find the best set of parameters for the net to extract the proper information, an hyperparameter tuning on each body sets order was done. Hyperparameters and test accuracies are shown in Table 4.3. Different parameters were found for different body sets orders. The net structure changed to properly extract the information from pose features.

Table 4.3: Hyperparameters and model test accuracies for each body set order.

Body set order	**Residual Blocks**	**Batch size**	**Optimizer**	**Number of filters**	**Kernel size**	**Test Accuracy (%)**
RTL	3	16	SGD	16	5	76
TLR	3	16	ADAM	16	5	80
LTR	5	16	SGD	8	3	83
RLT	3	16	SGD	16	5	81
TRL	3	16	SGD	16	3	85
LRT	3	16	SGD	16	5	85

Since Head and Trunk body set does not characterize gestures' kinematics (those keypoints have a smaller action volume with respect to arms body sets), RTL and LTR body sets orders are harder for the algorithm to learn. In fact, net's filters try to find local spatial correlations within images and it is better for the algorithm to have arms body sets adjacent, especially for symmetric arm gestures. This is true for all gestures, with the exception of *yes* and *no* gestures. Moreover, Head and Trunk, Right limb, Left limb (TRL) body segments' organisation and its opposite Left limb, Right limb, Head and Trunk (LRT) turned out to be the best ones for the algorithm to learn, with a test accuracy of 85%. Indeed, asymmetric gestures were preformed with the right arm in Healthy Dataset and the central position of the right arm body set allowed net's filters to have a higher chance

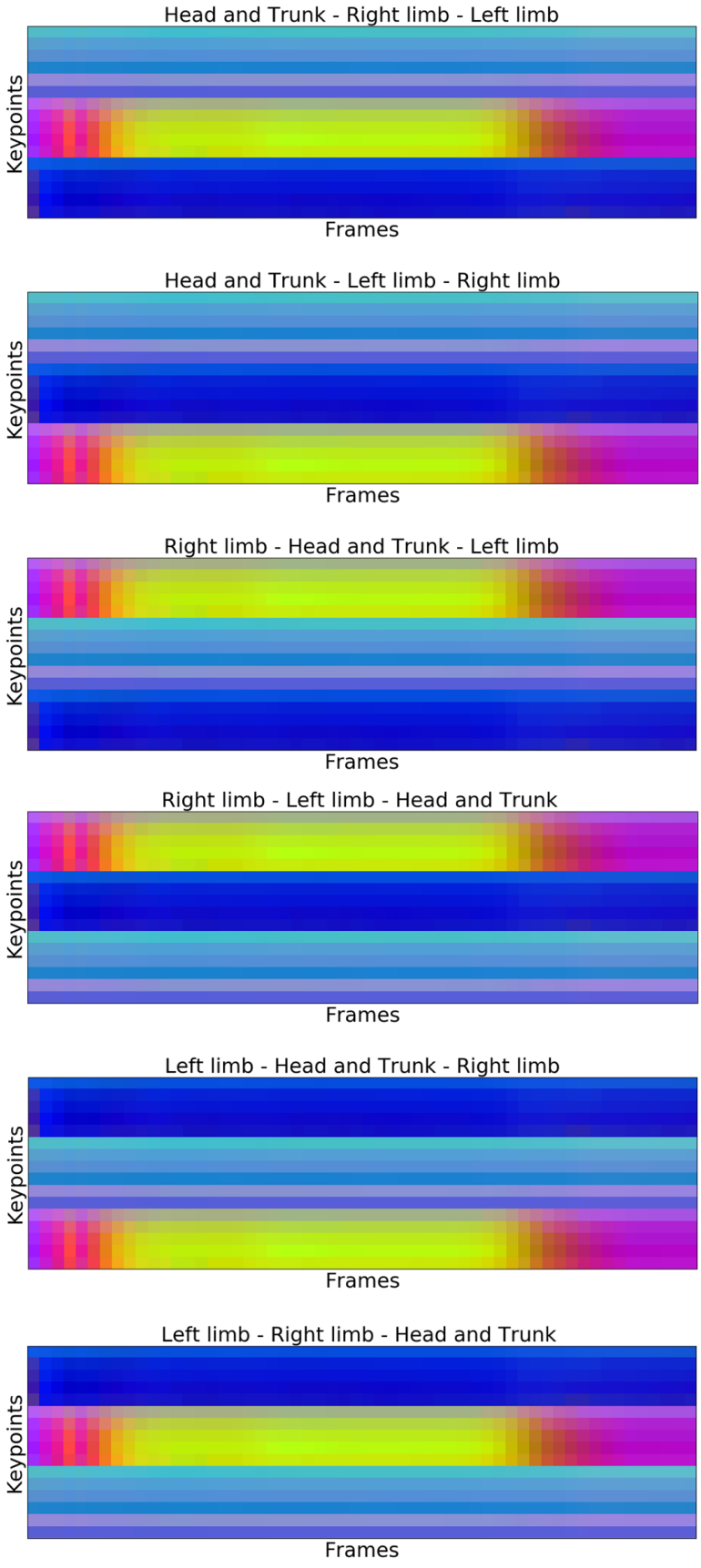

Figure 4.2: RGB pose features of pointing gestures with different body set orders. Keypoints on vertical axis while the total number of frames on the horizontal one.

to extract more information. In this way, before reaching the edges of the images, sliding filters passed over the characterizing arm several times, promoting network learning of the gestures.

4.2.2 Gestures Normalizations

Once the best keypoints' arrangement was found, this work focused on normalizing skeletons' coordinates. Table 4.4 shows results of the different experiments carried out and described in Subsection 3.4.3. The best result was achieved with the *gesture-independent*

Table 4.4: Comparison of different Coordinates Normalizations' combinations on the basis of Test Accuracy.

Train and Validation	**Test**		**Test Accuracy (%)**
gesture-independent	gesture-independent	per body control volume	74
		per keypoint	51
gesture-dependent	gesture-independent	per body control volume	38
		per keypoint	11

normalization both for training/validation and testing, over the whole-body control volume. *Gesture-dependent* normalization should have provided better user-invariant results thanks to the definition of an action-specific displacement range equal for all subjects. However, in test phase this approach could not be applied because gestures had not been recognized yet. Therefore, *gesture-independent* normalization turned out to be the best choice because, to this extent, both train/validation and test *pose features* were normalized in the same way. The distances computed to analyse the difference between *gesture-dependent* and *gesture-independent* normalizations were compared for each gesture class and results are shown in the chart of Figure 4.3.

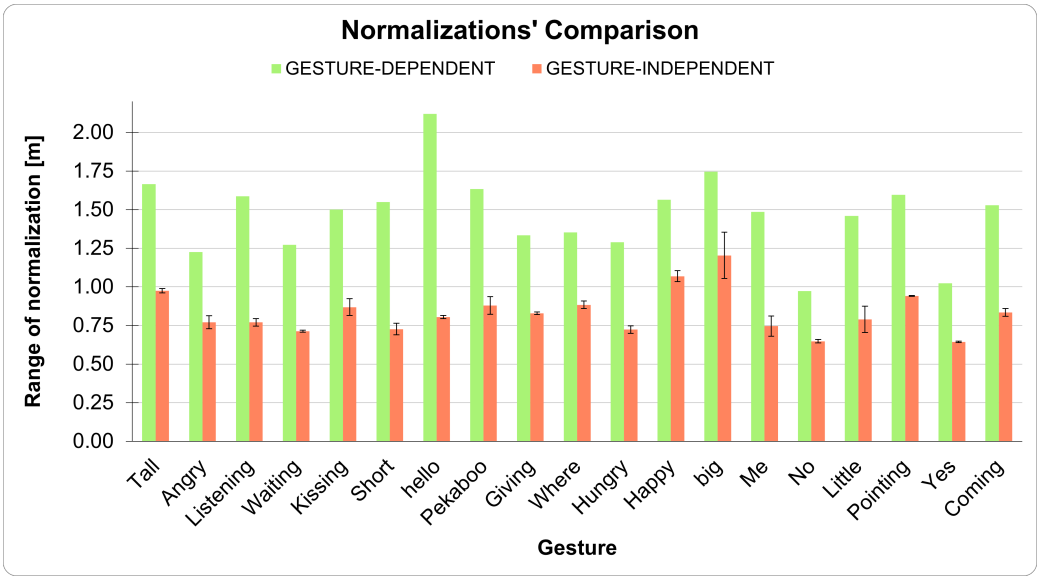

Figure 4.3: The chart shows how the gesture-dependent and gesture-independent normalizations are characterized by different ranges of normalization. The range of normalization was defined by Equation 3.14 for gesture-dependent normalization and Equation 3.15 for gesture-independent normalization.

Note that in *gesture-independent* normalization, each sample had its own normalization's range, thus, for the comparison, a mean per gesture class was plotted together with error bars. The graph shows how the *gesture-dependent* and *gesture-independent* normalizations are characterized by different ranges of normalization. For this reasons, it's difficult for the model to "understand" test images different from training ones and lower test accuracies were obtained in these cases.

Moreover, as expected, *per body control volume* normalization turned out to be better with respect to *per keypoint* one. *Per keypoint* normalization defines as many control volumes as keypoints' number, outlining each keypoint's kinematics range. To recognize actions, the net needs to find correlations between keypoints, thus coordinates must be scaled in the same range. *Per body control volume* normalization defines a single control volume for all keypoints, referring each keypoint to the same range. For this reasons *per body control volume* normalization turned out to be the best one.

4.2.3 Temporal Interpolation and Reshape

In Figure 4.4, a *peekaboo* RGB pose feature is shown before the temporal interpolation with the whole temporal dimension on the horizontal axis. Figure 4.5 (*left*) illustrates the RGB pose of the gesture after the temporal interpolation, with a 48×48 resolution and keypoints stacked three times ($aaa - bbb - ccc$ configuration). Figure 4.5 (*right*) shows the RGB pose after the temporal interpolation, with a 48×48 resolution, tripling the RGB pose feature vertically ($abc - abc - abc$ configuration).

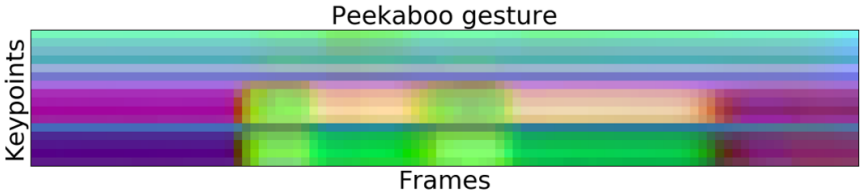

Figure 4.4: RGB Pose feature before temporal interpolation.

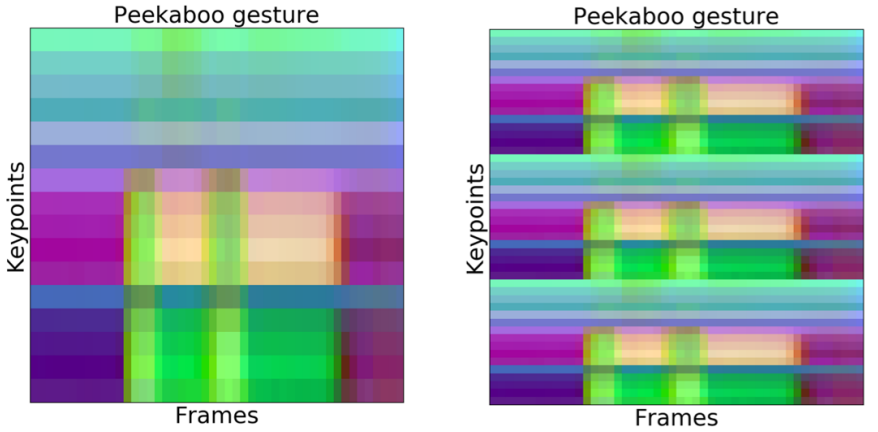

Figure 4.5: Left: $aaa - bbb - ccc$ keypoint configuration of RGB peekaboo pose feature; Right: $abc - abc - abc$ keypoint configuration of RGB peekaboo pose feature.

Tests' results are shown in Table 4.5. $abc - abc - abc$ pose feature configuration turned out to be the best for the algorithm to learn with a test accuracy of 95 % with respect to 87 % of the other configuration. Tests were carried out with a kernel size of 5 (resulted from net's hyperparameters tuning). With $aaa - bbb - ccc$ pose feature configuration each joint dynamics is tripled and thus emphasized. The net extracts information from a narrow region of the pose feature relative to only two keypoints at a time. In $abc - abc - abc$ pose feature configuration, the action is tripled and the net's filters analyze more keypoints at a time. As a gesture is defined by the relative position between keypoints, the net has to focus on more keypoints. This is possibile with the $abc - abc - abc$ pose feature configuration. With a larger kernel size extracting information on $aaa - bbb - ccc$ pose features, the sliding filter would have analyzed more keypoints but a lower number of times. Thus, $abc - abc - abc$ pose configuration would have been the best one even in this case.

Table 4.5: Parameters and model test accuracies for $aaa - bbb - ccc$ pose feature configuration and $abc - abc - abc$ pose feature configuration

Configuration	**Residual Blocks**	**Batch**	**Optimizer**	**Filters**	**Kernel size**	**Test Accuracy (%)**
$aaa - bbb - ccc$	5	16	SGD	16	5	87
$abc - abc - abc$	5	8	SGD	8	5	95

4.2.4 Enhanced Action Images

In order to highlight each RGB pose feature, contrast of images was enhanced exploiting CLAHE. Figure 4.6 shows *hello* RGB pose feature before and after CLAHE application while in Figure 4.7 pose features' histograms are illustrated. The original histogram has been stretched to the far ends and equalized.

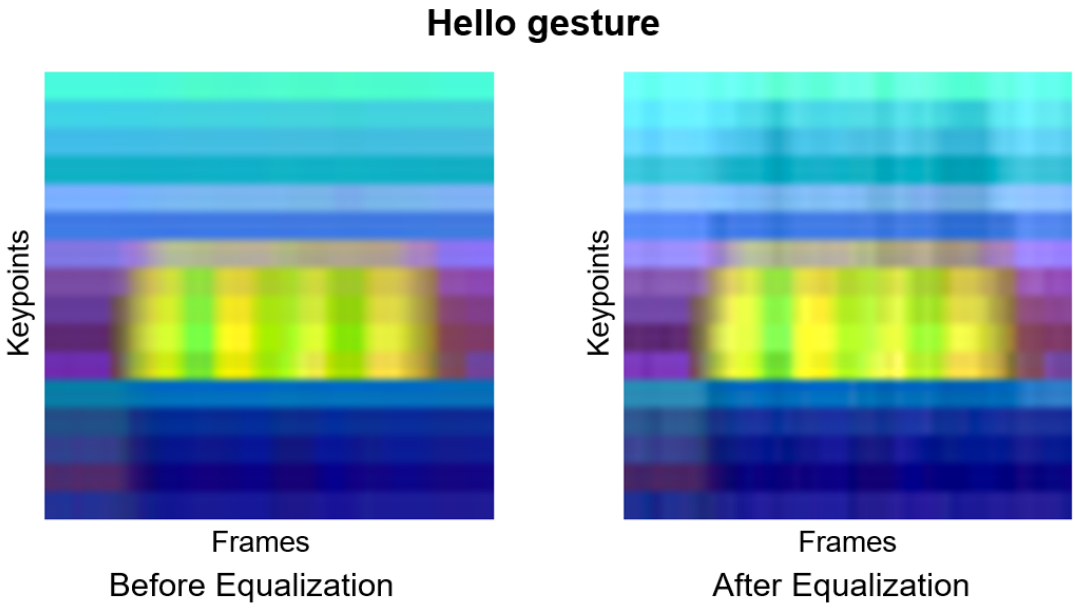

Figure 4.6: Before and after CLAHE on "hello" RGB pose features.

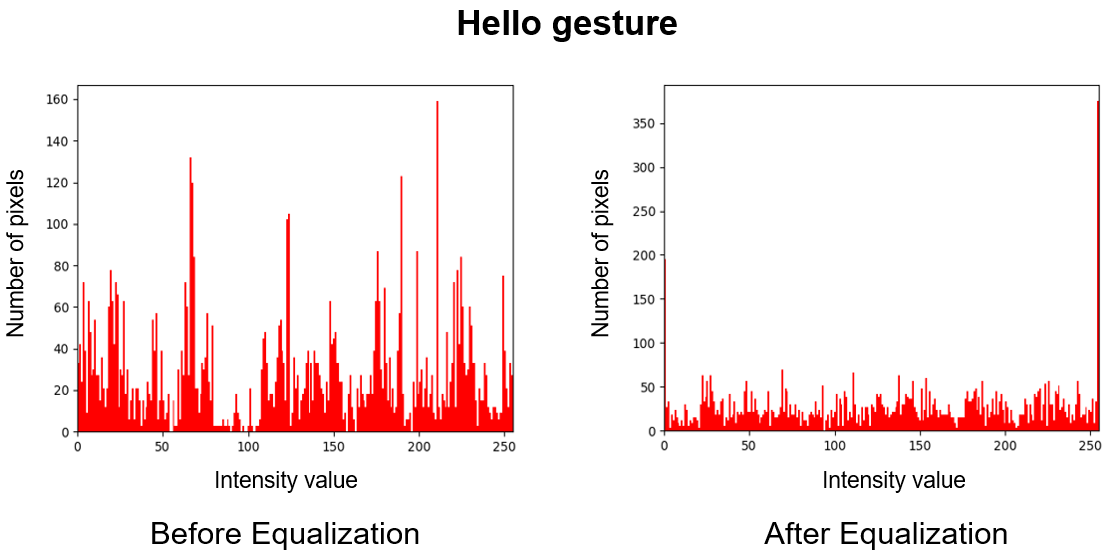

Figure 4.7: Histograms before and after CLAHE on "hello" RGB pose features.

After CLAHE application, local contrast is enhanced thus emphasizing the movements of each pose feature from the biomechanical point of view. This allows the net to better recognize a gesture.

4.2.5 Data Mirroring

In Data Mirroring technique, the x coordinates of each frame in each pose feature were mirrored with respect to the *Reference Point*. In Figure 4.8 a *kissing* gesture RGB pose feature before and after Data Mirroring is shown in TRL body set configuration ($aaa-bbb-ccc$ configuration for better understanding). As can be seen, the gesture was performed

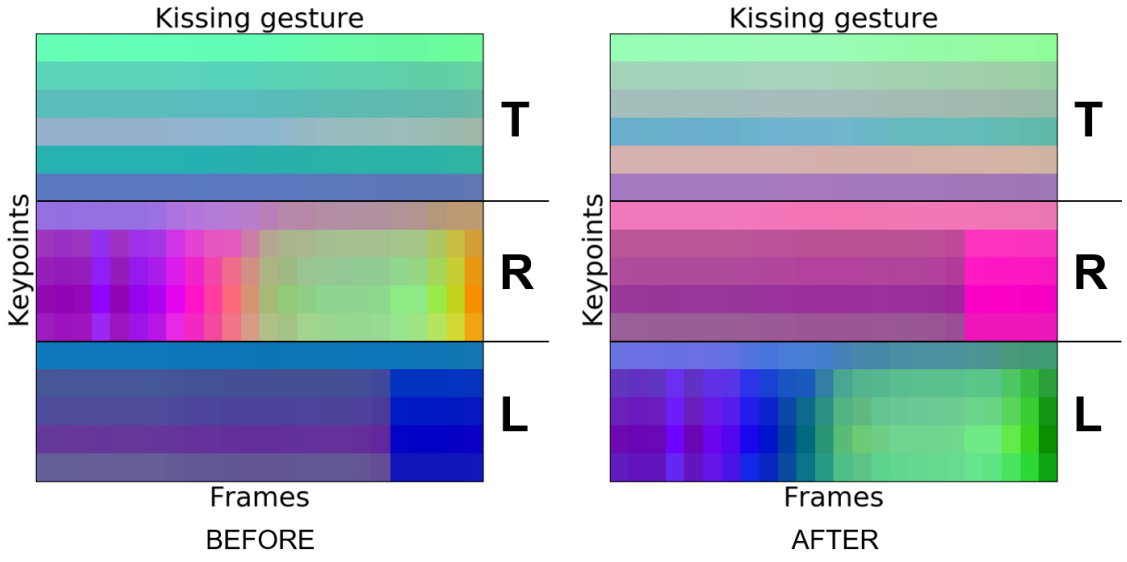

Figure 4.8: Kissing gesture RGB pose feature before and after Data Mirroring, TRL body set configuration, $aaa - bbb - ccc$ configuration

with the right hand. After Data Mirroring, all coordinates are mirrored, but the relevant color changing (thus, displacement) involved the Right limb body set only. In fact, the color changing from purple/pink to yellow in the RGB pose feature, corresponding to the right limb's displacement in doing the gesture, is now depicted in the Left limb body set with a color changing from purple/blue to green. This led the net to recognize left-handed gestures. In Figure 4.9 a *big* gesture RGB pose feature before and after Data Mirroring is shown in TRL body set configuration ($aaa - bbb - ccc$ configuration for better understanding). In this case, since the action was symmetric, the mirroring technique

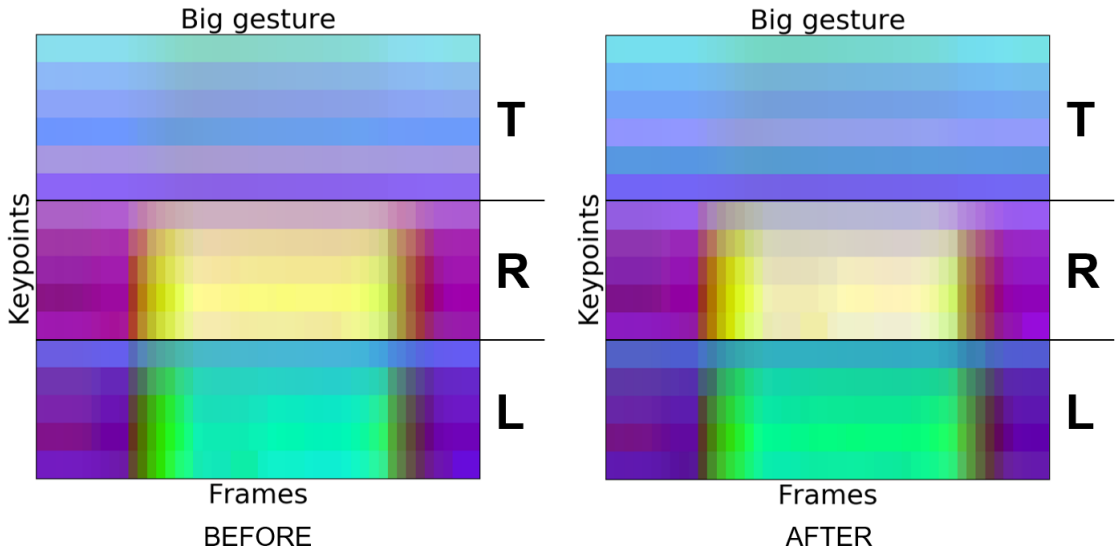

Figure 4.9: Big gesture RGB pose feature before and after Data Mirroring, TRL body set configuration, $aaa - bbb - ccc$ configuration

resulted in data augmentation. In fact, displacement (thus, color changing) involved both right arm and left arm. A new sample was added, thus the dataset increased.

4.3 Classification

4.3.1 Hyperparameters Tuning

Net's hyperparameters have been set after exploiting Ax platform in order to better tune the algorithm. Parameters' sets for Subsampled Healthy Dataset, Healthy Dataset and Expanded Dataset are shown in the following Table 4.6.

Table 4.6: Net's hyperparameters resulted from the tuning for Subsampled Healthy Dataset, Healthy Dataset and Expanded Dataset.

Hyperparameters	**Subsampled**	**Healthy**	**Expanded**
Number of Res Blocks	3	3	5
Batch size	8	8	8
Number of Filters	-	4	8
Kernel Size of Filters	-	5	3

For what concerns the number of residual blocks (Res Blocks), Subsampled and Healthy datasets had 3 of them while the Expanded Dataset was characterized by 5 of them. This result was due to the need of a deeper neural network able to extract more information for the Expanded Dataset's wider gesture set.

Batch size could range between [8, 16, 32, 64, 256]. All datasets had a batch size of 8 meaning that in one training epoch, net's weights were updated $n/8$ times, with n as the number of samples in the dataset. This small batch size allowed the model to be more generalized because it was updated more times in a train epoch with respect to a bigger batch size.

Number of filters and Kernel size involved Healthy and Expanded datasets only. With Healthy Dataset, net's number of filters was lower with respect to the Expanded Dataset since the first dataset inputs' size had a lower resolution (32×32 with respect to 48×48), thus a lower information content. On the other hand, Kernel Size was bigger with respect to the Expanded Dataset beacuse of the pose features' configuration ($aaa - bbb - ccc$ vs $abc - abc - abc$ configuration).

4.3.2 Offline Models

In this Subsection all the offline models relative to Subsampled Healthy Dataset, Healthy Dataset and Expanded Dataset are presented.

5-gestures Model

Since Subsampled Healthy Dataset was used as first approach to gesture recognition algorithms, basic data processing was exploited. All data transformations are shown in Table 4.7. Data mirroring was not yet implemented, so the algorithm did not recognize left-handed gestures.

Table 4.7: Subsampled Healthy Dataset Data Transformations.

Reference Keypoint	**Normalization Value**	**Gesture Normalization**	**Resize**	**Data Mirroring**	**Enhanced Action Images**
Shoulder Center	Total height	gesture-independent per body control volume	32×32	No	No

With these data transformations and with a 5 gestures set, the model reached **94%** test accuracy. As can be seen from the confusion matrix in Figure 4.10, almost all gestures are correctly predicted by the model. Some confusions could be due to ordinary inter-subjects variability in performing gestures. In Table 4.8 metrics scores are shown. *pointing* and *little* gestures, when detected by the algorithm, are always recognized (precision 100%), while for *coming* gesture, the algorithm correctly identifies 100% of all that kind of gesture (recall 100%).

14-gestures Model 1

The same Subsampled Healthy Dataset's data transformation was applied to Healthy Dataset with a 14 gesture set (Table 4.9). Data Mirroring was still not implemented, so this model was not able to recognize left-handed gestures.

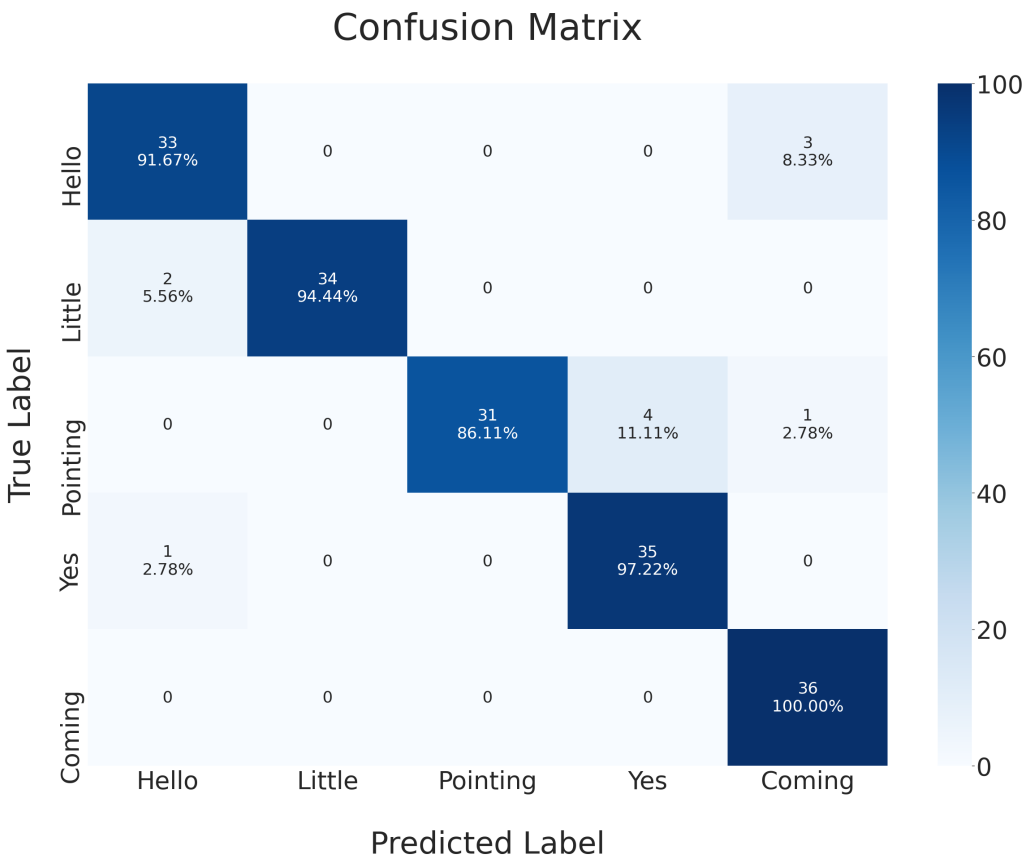

Figure 4.10: Confusion Matrix of the 5-gestures model.

Table 4.8: Metrics scores of 5-gestures model.

Gesture (%)	**Accuracy (%)**	**F1-Score (%)**	**Precision (%)**	**Recall (%)**
Hello	97	92	92	92
Little	99	97	100	94
Pointing	97	93	100	86
Yes	97	93	90	97
Coming	98	95	90	100

Table 4.9: Healthy Dataset Data Transformations 1.

Reference Keypoint	**Normalization Value**	**Coordinate Normalization**	**Resize**	**Data Mirroring**	**Enhanced Action Images**
Shoulder Center	Total height	gesture-independent per body control volume	32×32	No	No

As expected, with a wider gesture set, results were worse than the previous. The model reached a test accuracy of **78%**. Confusion matrix and metrics scores are shown in Figure 4.11 and Table 4.10 respectively. Note that *big* gesture was mistaken for the *tall* gesture. This is due to the way the subject was performing the gesture in data acquisitions. For this reason F1-score and Precision of *big* gesture were not computed and Recall was 0 while *tall* gesture had the lowest Precision value (43%). *Hello* Precision was 50%, meaning that only half of gestures predicted as *hello* were actually part of this gesture class. *Where*, *short*, *pointing*, *hungry* and *coming* gestures had the highest metric scores (100%).

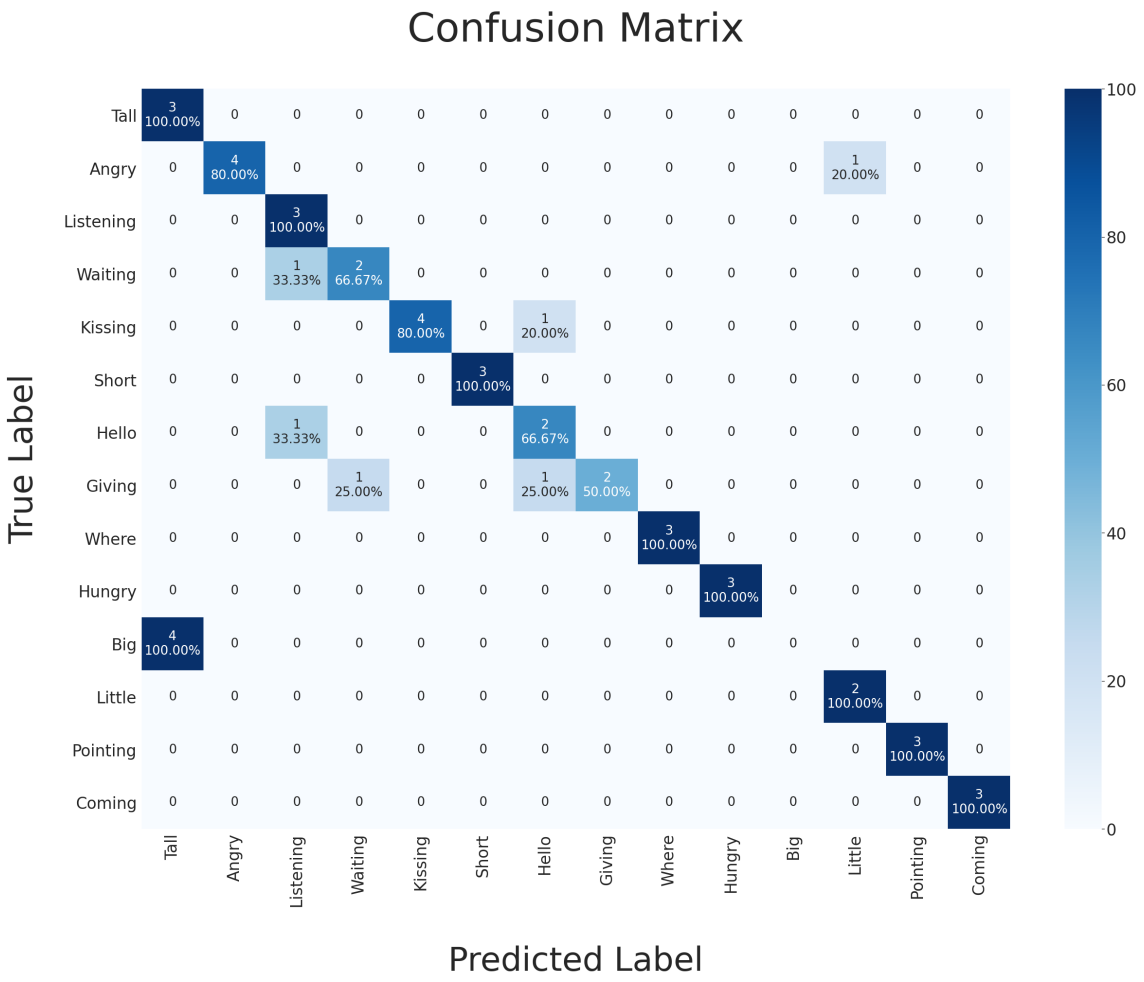

Figure 4.11: Confusion Matrix of 14-gestures Model 1.

14-gestures Model 2

Since trunk size turned out to be the most stable and less variant normalization, as mention in Subsection 4.1.1, a new model was trained exploiting this data transformation (Table 4.11). The new model reached an higher test accuracy: **85%**. Confusion matrix and metrics scores are shown in Figure 4.12 and Table 4.12 respectively. Note that *waiting*, *kissing* and *coming* gestures have the lowest recall (33% 60% 33% respectively), since they have similar action range in space with respect to other gestures.

Table 4.10: Metrics scores of the 14-gestures Model 1.

Gesture	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
Tall	90	60	43	100
Angry	97	89	100	80
Listening	95	75	60	100
Waiting	95	67	67	67
Kissing	97	89	100	80
Short	100	100	100	100
Hello	93	57	50	67
Giving	95	67	100	50
Where	100	100	100	100
Hungry	100	100	100	100
Big	90	nan	nan	0
Little	97	80	67	100
Pointing	100	100	100	100
Coming	100	100	100	100

Table 4.11: Healthy Dataset Data Transformations 2.

Reference Keypoint	Normalization Value	Gesture Normalization	Resize	Data Mirroring	Enhanced Action Images
Shoulder Center	Trunk size	gesture-independent per body control volume	32×32	No	No

Table 4.12: Metrics scores of the 14-gestures Model 2.

Gesture	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
Tall	100	100	100	100
Angry	100	100	100	100
Listening	93	67	50	100
Waiting	95	50	100	33
Kissing	93	67	75	60
Short	93	67	50	100
Hello	98	80	100	67
Giving	100	100	100	100
Where	100	100	100	100
Hungry	100	100	100	100
Big	100	100	100	100
Little	100	100	100	100
Pointing	100	100	100	100
Coming	95	50	100	33

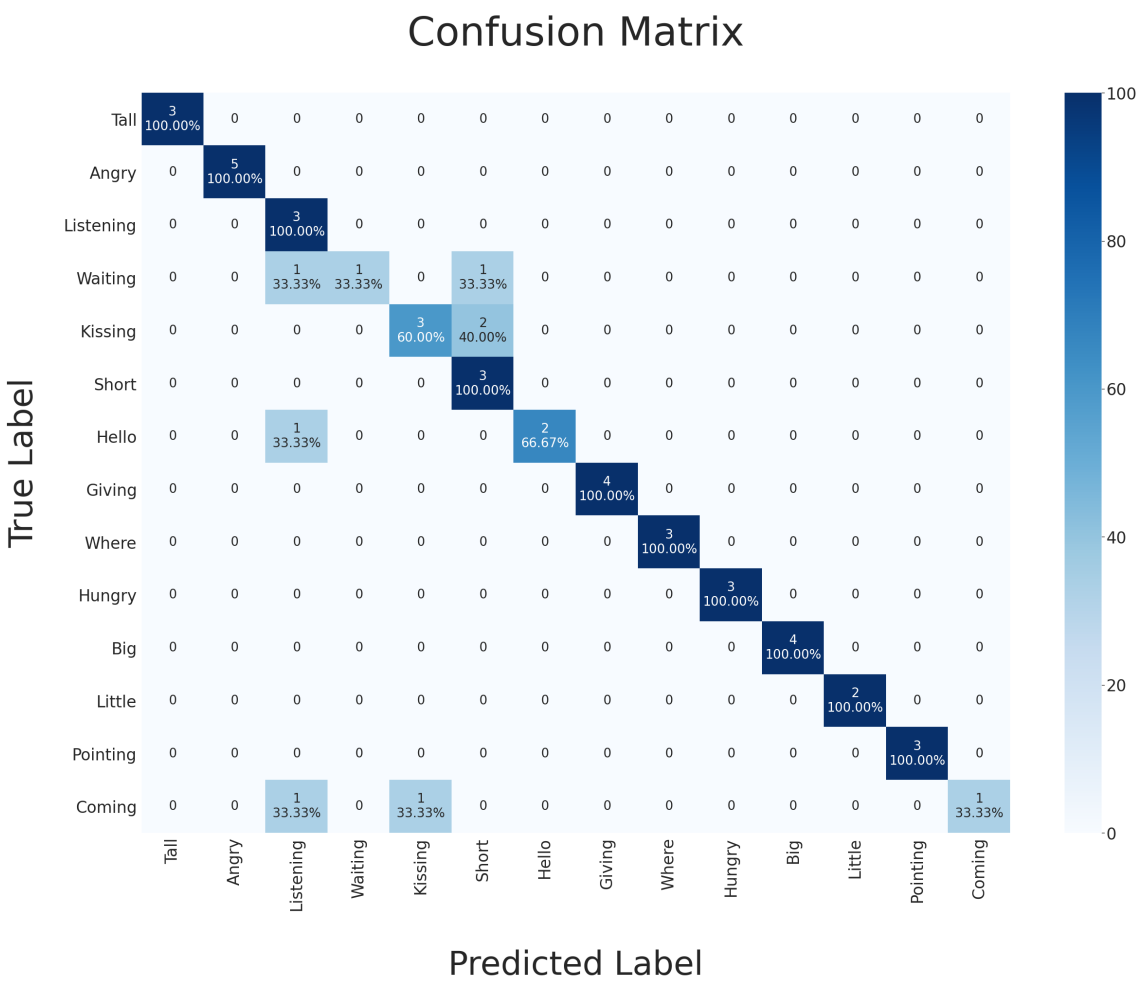

Figure 4.12: Confusion Matrix of the 14-gestures Model 2.

11-gestures Model

Gestures with similar action range in coordinate's space are difficult for the algorithm to distinguish. For this reason, some "tricky" gestures were deleted: *waiting*, *kissing* and *coming*. The same data transformation reported in the previous model was applied with no Data Mirroring (Table 4.13). Thus, the model could not recognize left-handed gestures yet.

Table 4.13: Healthy Dataset Data Transformations 2.

Reference Keypoint	Normalization Value	Gesture Normalization	Resize	Data Mirroring	Enhanced Action Images
Shoulder Center	Trunk size	gesture-independent per body control volume	32×32	No	No

The new model reached **97%** test accuracy. The confusion matrix is shown in Figure 4.13 while metrics scores in Table 4.14. As expected, recall and precision are way better now, but the gesture set is reduced.

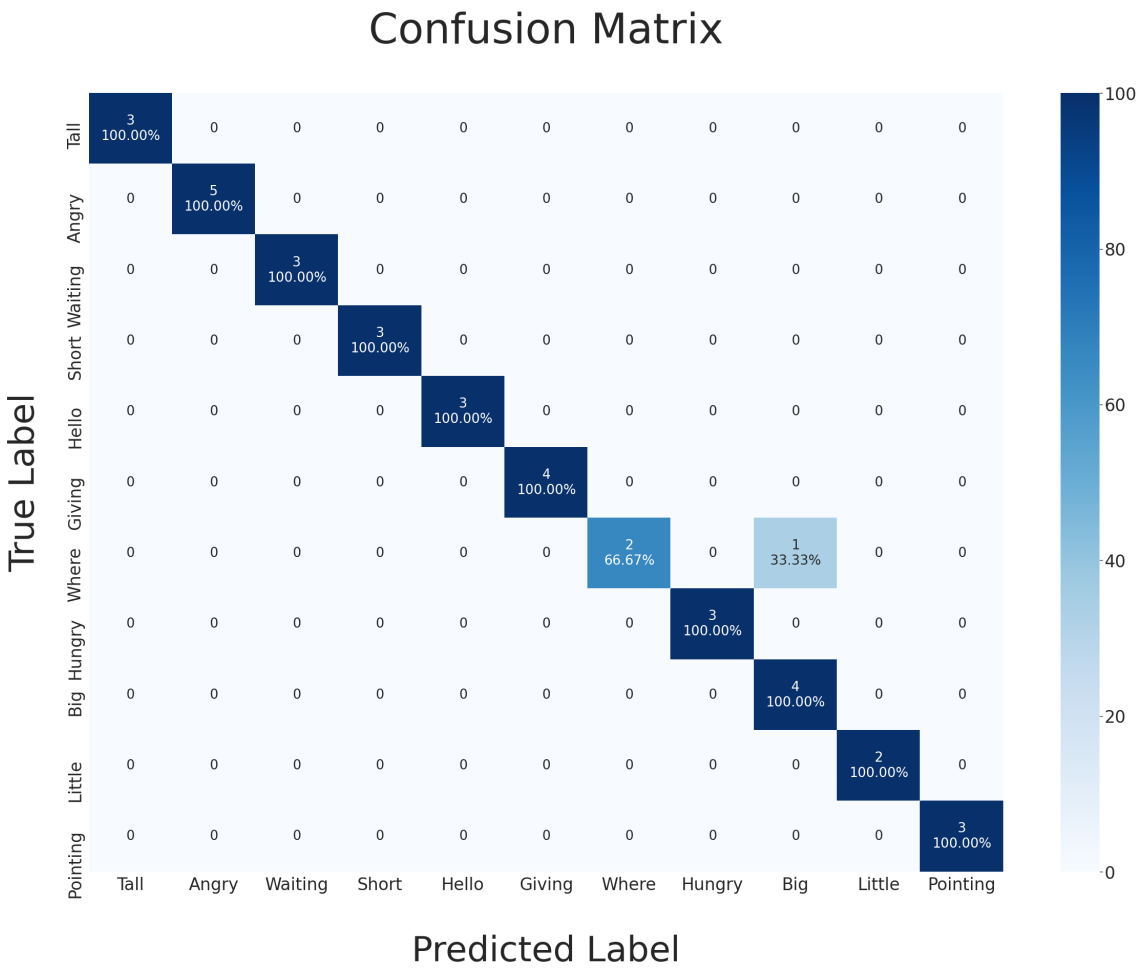

Figure 4.13: Confusion Matrix of 11-gestures Model.

Table 4.14: Metrics scores of 11-gestures Model.

Gesture	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
Tall	100	100	100	100
Angry	100	100	100	100
Waiting	100	100	100	100
Short	100	100	100	100
Hello	100	100	100	100
Giving	100	100	100	100
Where	97	80	100	67
Hungry	100	100	100	100
Big	97	89	80	100
Little	100	100	100	100
Pointing	100	100	100	100

19-gestures Model

With Expanded Dataset, new methods' settings were applied as summarized in Table 4.15.

Table 4.15: Expanded Dataset Data Transformations.

Reference Keypoint	Normalization Value	Gesture Normalization	Resize	Data Mirroring	Enhanced Action Images
Hip Center	Trunk size	gesture-independent per body control volume	48×48 *abc-abc-abc* configuration	Yes	Yes

As already mentioned in Subsection 4.1.1, hip center reference keypoint was chosen as the most stable option for frame by frame normalization. Moreover, a new reshape 48×48 pixels with the best pose feature configuration (*abc-abc-abc* configuration) and CLAHE were exploited for training, validating and testing the model. Since Data Mirroring was included, the model could recognize actions independently of the dominant hand. With all the 19 gestures a test accuracy of **95%** was reached. Confusion matrix and metrics scores are shown in Figure 4.14 and Table 4.16 respectively. Note that some gestures were still mistaken for others even after implementing all data processing's best results. In particular, actions like *short* or *giving* are similar gestures which can be easily mistaken if performed by different subjects. Considering the wide gesture set and the different temporal dynamics and duration of actions, results were encouraging in sight of online recognition.

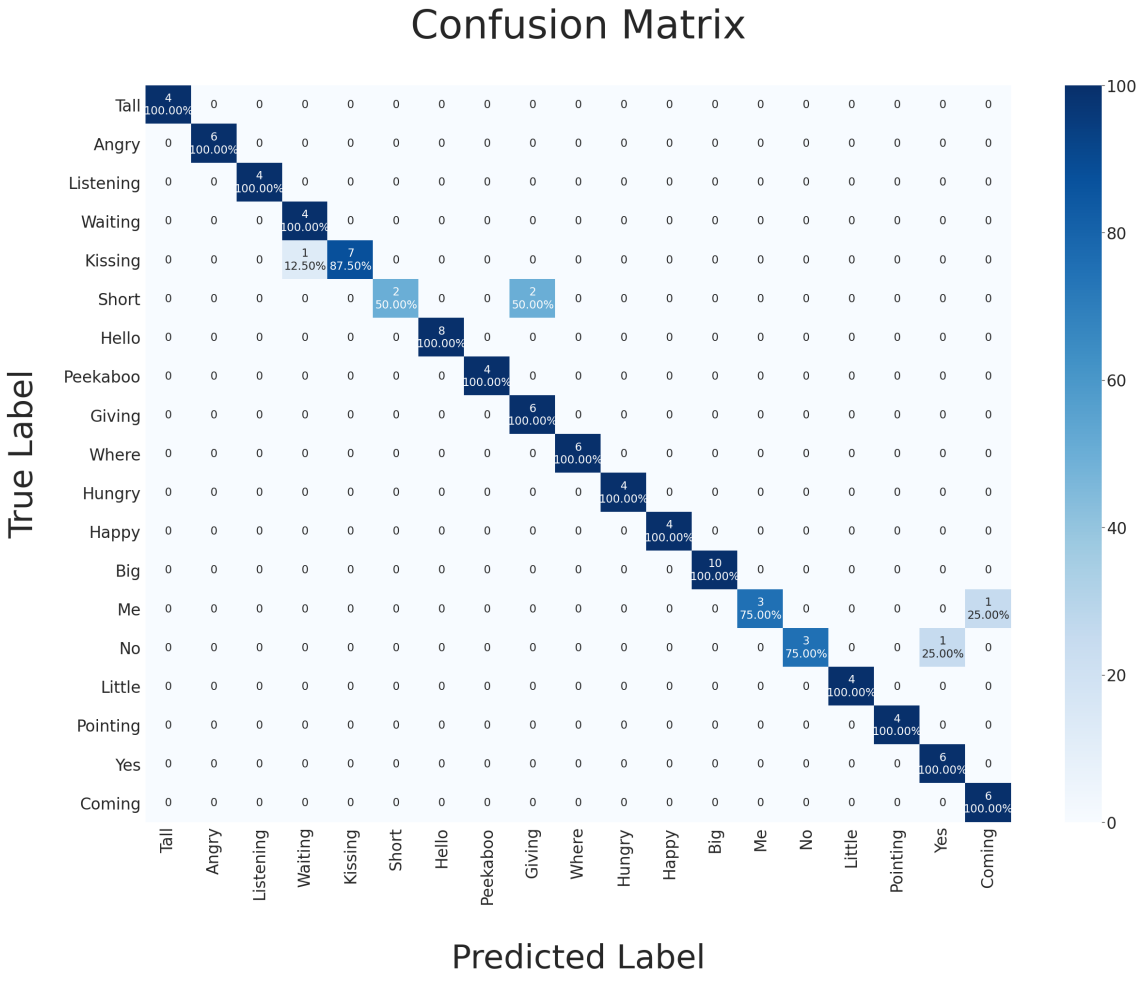

Figure 4.14: Confusion Matrix of the Expanded Dataset Model with 19 gestures.

Table 4.16: Metrics scores of 19-gestures Model.

Gesture	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
Tall	100	100	100	100
Angry	100	100	100	100
Listening	100	100	100	100
Waiting	99	89	80	100
Kissing	99	93	100	88
Short	98	67	100	50
Hello	100	100	100	100
Peekaboo	100	100	100	100
Giving	98	86	75	100
Where	100	100	100	100
Hungry	100	100	100	100
Happy	100	100	100	100
Big	100	100	100	100
Me	99	86	100	75
No	99	86	100	75
Little	100	100	100	100
Pointing	100	100	100	100
Yes	99	92	86	100
Coming	99	92	86	100

4.4 Online Recognition

Online final settings for Kinect-only and Kinect-NAO configurations with 11-gestures Model and 19-gestures Model are presented in this section. 11-gestures Model was used for the first set of new acquisitions. When 19-gestures Model was developed, it was used for more recent clinical applications and online testing. Thus, the two models were both set online with different parameters' configurations. Considering the wider gesture set involved, parameters' choice for 19-gestures Model turned out to be more challenging with respect to 11-gestures Model.

First, to undertand the difference between Kinect-only and Kinect-NAO configurations, Kinect sampling frequency was recorded. In Figure 4.15 sampling frequency variations along time are shown and in Table 4.17 Frames Per Second (FPS) mean and variance are computed for both configurations.

Table 4.17: Frames per second mean and variance for Kinect-only and Kinect-NAO configuration.

Configuration	FPS mean	FPS variance
Kinect-only	50.48	14.65
Kinect-NAO	11	3.56

As it can be seen, Kinect-NAO configuration slowed down the frames' capture by the camera. When performing gestures in front of Kinect camera, Kinect-only FPS's mean value was 50.48 fps while, with robot connection, the mean value decreased to 11 fps. A lower FPS means that an action is described by a lower number of captured frames, thus reducing the information content. This behaviour had to be taken into account since the

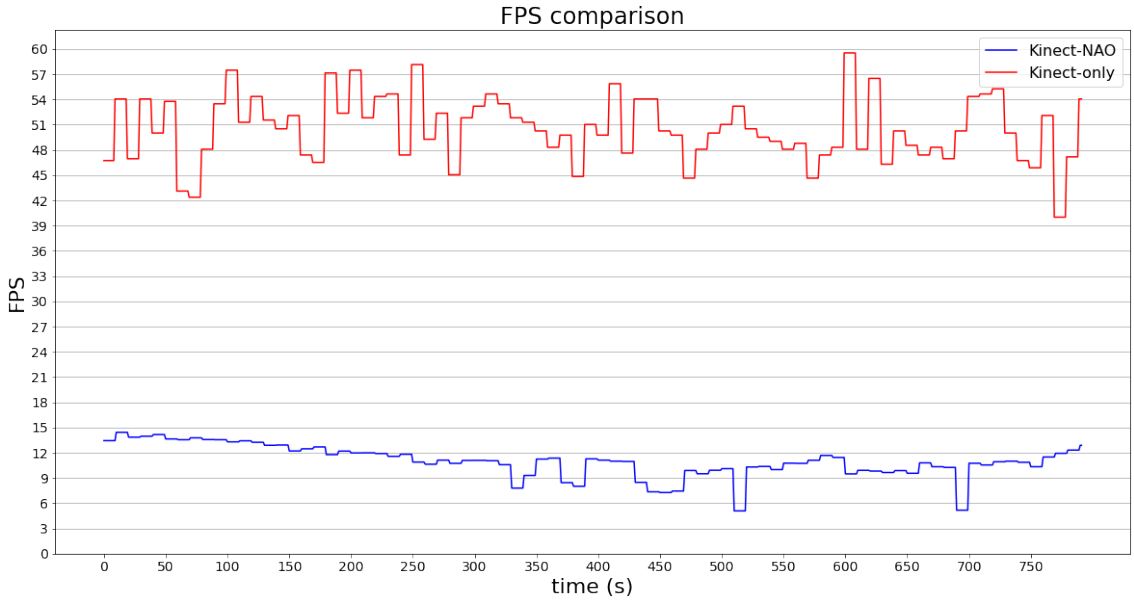

Figure 4.15: FPS comparison: Kinect-only and Kinect-NAO configurations.

recognition task relies on data acquisition of skeleton poses.

4.4.1 Kinect-only Settings

In order to exploit the recognition algorithm in a real-time classification, a sliding window was used: the *size* was set to 68 frames and computed as a mean value. Particularly, for each gesture class in the dataset the average number of frames was calculated as shown in Figure 4.16. In order to set window *size* the mean value over all gestures was estimated. Number of frames' mean values and standard deviations for each gesture are shown in Table 4.18. Note that the same gesture could be performed in a longer or shorter period

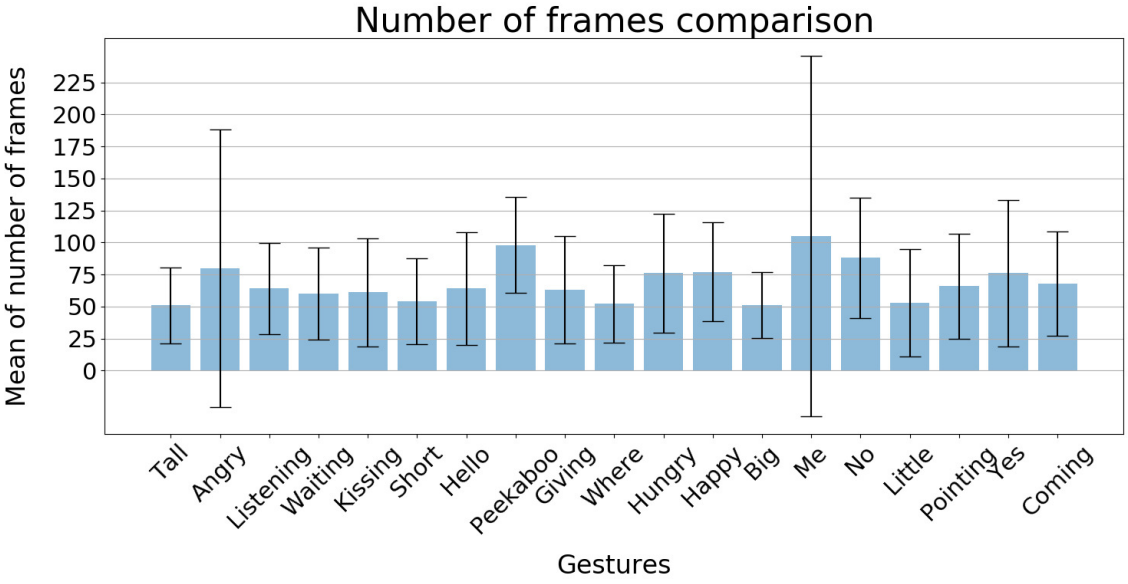

Figure 4.16: Frames comparison for each gesture.

of time. For this reason, gestures like *me* and *angry* have a high standard deviation.

Table 4.18: Number of frames' mean and standard deviation for each gesture. To set the sliding window's size, the overall mean was computed.

Gestures	**Mean number of frames**	**Std**
Tall	51	29
Angry	80	108
Listening	64	35
Waiting	60	36
Kissing	61	42
Short	54	33
Hello	64	43
Peekaboo	98	37
Giving	63	41
Where	52	30
Hungry	76	46
Happy	77	38
Big	51	25
Me	105	140
No	88	46
Little	53	42
Pointing	66	41
Yes	76	57
Coming	68	40
Overall Mean	**68**	

Threshold τ was set to 0.55, which is the minimum probability for the algorithm to recognize correctly a gesture. All minimum probabilities in the gesture set in Expanded Dataset are shown in Table 4.19.

Since some gestures can be identified even when another gesture is performed (for instance, *short* gesture when doing *giving* gesture), gesture-specific thresholds were also set. When the detection threshold was exceeded, the probabilities' prediction vector was saved in a *buffer*. Then, the window slided of a fixed *step* before predicting again. Different *step* values in terms of number of frames and different implementations were experimented for 11-gestures Model and 19-gestures Model.

11-gestures Model

A single frame *step* was tested for the first online recognition implementation. Therefore, when the buffer was filled with N prediction vectors, the algorithm checked whether they were equal. Different buffer's length were experimented: N=4 probabilities' prediction vectors turned out to be the best choice.

Table 4.19: Minimum softmax probabilities to get a true positive in the gesture set.

Gesture	**Minimum probabilities**
Tall	0.99
Angry	0.74
Listening	0.51
Waiting	0.60
Kissing	0.64
Short	**0.55**
Hello	0.70
Peekaboo	0.97
Giving	0.54
Where	0.99
Hungry	0.95
Happy	0.60
Big	0.64
Me	0.86
No	0.64
Little	0.80
Pointing	0.99
Yes	0.89
Coming	0.69

19-gestures Model

A 30 frames *step* was proved to be the best choice for the algorithm to predict the gesture. In fact, if the window's *step* is set to 1, consecutive pose features are similar because the action kinematics does not actually change due to high Kinect's FPS. Once the buffer was filled with 8 prediction vectors, the algorithm identified the gesture performed by averaging *buffer*'s prediction vectors' probabilities. Different buffer's length were experimented: N=8 probabilities' prediction vectors turned out to be the best choice in order to detect gestures with a long action kinematics in terms of time (for instance *happy* gesture) but even with a short one (for example *kissing* gesture). Finally, the algorithm identified the gesture performed by averaging *buffer*'s prediction vectors' probabilities.

4.4.2 Kinect-NAO Settings

Since Kinect's performance changed with NAO connection as mentioned before, new parameters were tested.

11-gestures Model

The first online implementation with NAO implied a further buffer with respect to Kinect-only implementation: the *memory buffer*. In fact, to be sure that predictions of the algorithm were relative to the action performed within the protocol's timings, the *memory buffer* collected 3 output predictions. Only if all of them were equal, the gesture was properly identified to trigger NAO's feedback.

19-gestures Model

The lower Kinect's sampling frequency led to set a shorter window *size*: 40 frames with respect to 68 in Kinect-only configuration. A larger window *size* would have stored more than the gesture performed, slowing the recognition task. For these reasons, a single frame *step* was implemented, with the window sliding over the continuous data stream captured by the camera. Buffer's length was preserved: N=8 predictions probabilities vectors were saved and the output prediction was given by their averaging. This configuration turned out to be the best for Kinect-NAO acquisitions in order to face the changing FPS problem. It has to be noted that other setting configurations were implemented before achieving the final results.

4.5 Acquisitions

New acquisitions were carried out both at Politecnico di Milano and at CARElab in Fondazione Don Gnocchi with the models presented in Subsection Offline Models.

4.5.1 @Politecnico Acquisitions

Kinect and Robot gesture recognition performances were analyzed through different acquisitions on two healthy subjects with the best model 19-gestures Model and its final online setup. *Yes* and *no* gestures were "silenced" even if the offline 19-gestures Model

properly recognized them with a 99% accuracy both. These gestures are quite challenging movements for the Kinect to capture. In fact, their characterizing movements are described by a small number of joints and a reduced motion range (they involve only the head region) and would need a finer tracking system to be correctly tracked only when intentionally performed. For all these reasons, sometimes the recognition time slowed down: *yes* and *no* predictions were discarded while doing other actions but not removed, since the algorithm still identified them.

Before starting with Kinect-only acquisitions, to analyze prediction's vector probabilities' trend, *tall*, *hello* and *little* gestures where performed in Kinect-only configuration (Figure 4.17). As expected, when the movements were performed, the gesture-corresponding

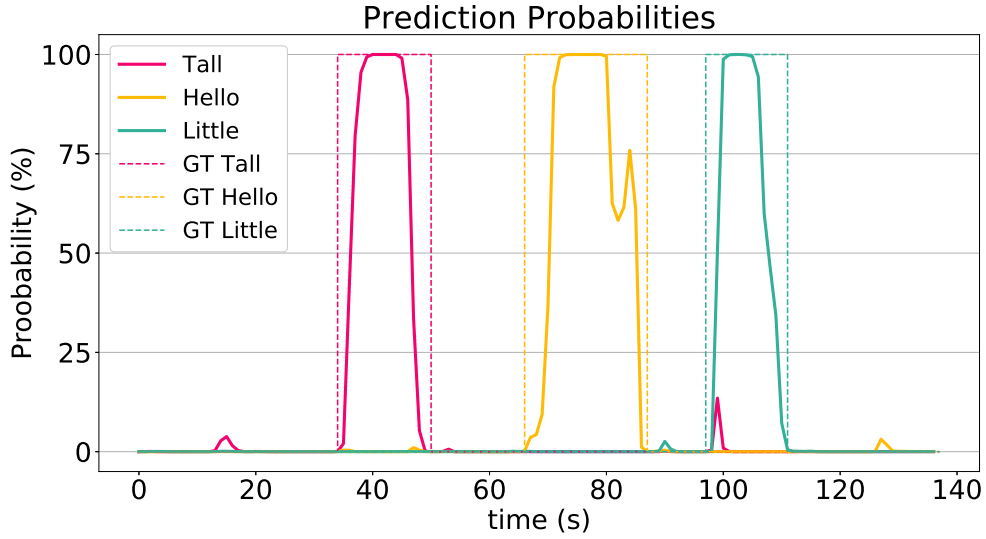

Figure 4.17: Probability trends of the prediction vector when performing tall, hello and little gestures. The step function stands for Ground Truth (GT) i.e. the temporal window in which the gesture was performed.

probability increased. Note that in *little* gesture, since at the beginning of the action both limbs are raised as in *tall* gesture, *tall* probability increased too.

Then, Kinect-only acquisitions were carried out. The confusion matrix is shown in Figure 4.18. *Angry* gesture was confused with *where*. This could be due to the fact that sometimes skeletal joints are not properly captured by Kinect camera. 17 gestures Accuracies, F1-scores, Precisions and Recalls are shown in Table 4.20. *Angry* gesture had a Recall of 50% because 2 out of 4 actions were confused, for this reason *where* gesture had a lower precision (67%). Kinect-only acquisitions reached an overall accuracy of **97%** and an F1 score of **97%** as shown in Table 4.21.

Then, Kinect-NAO acquisitions for the selected 17 gestures were carried out. The confusion matrix is shown in Figure 4.19. 17 gestures Accuracies, F1-scores, Precisions and Recalls are shown in Table 4.22. The confusion matrix and metrics scores show that *waiting* gesture was confused with *giving*: in fact, the two movements have a similar action volume range. For what concern *kissing* gesture mistaken with *happy*, keypoints' files were plotted mimicking skeleton movements. It turned out that joints were captured by Kinect in a wrong position, similar to *happy* gesture.

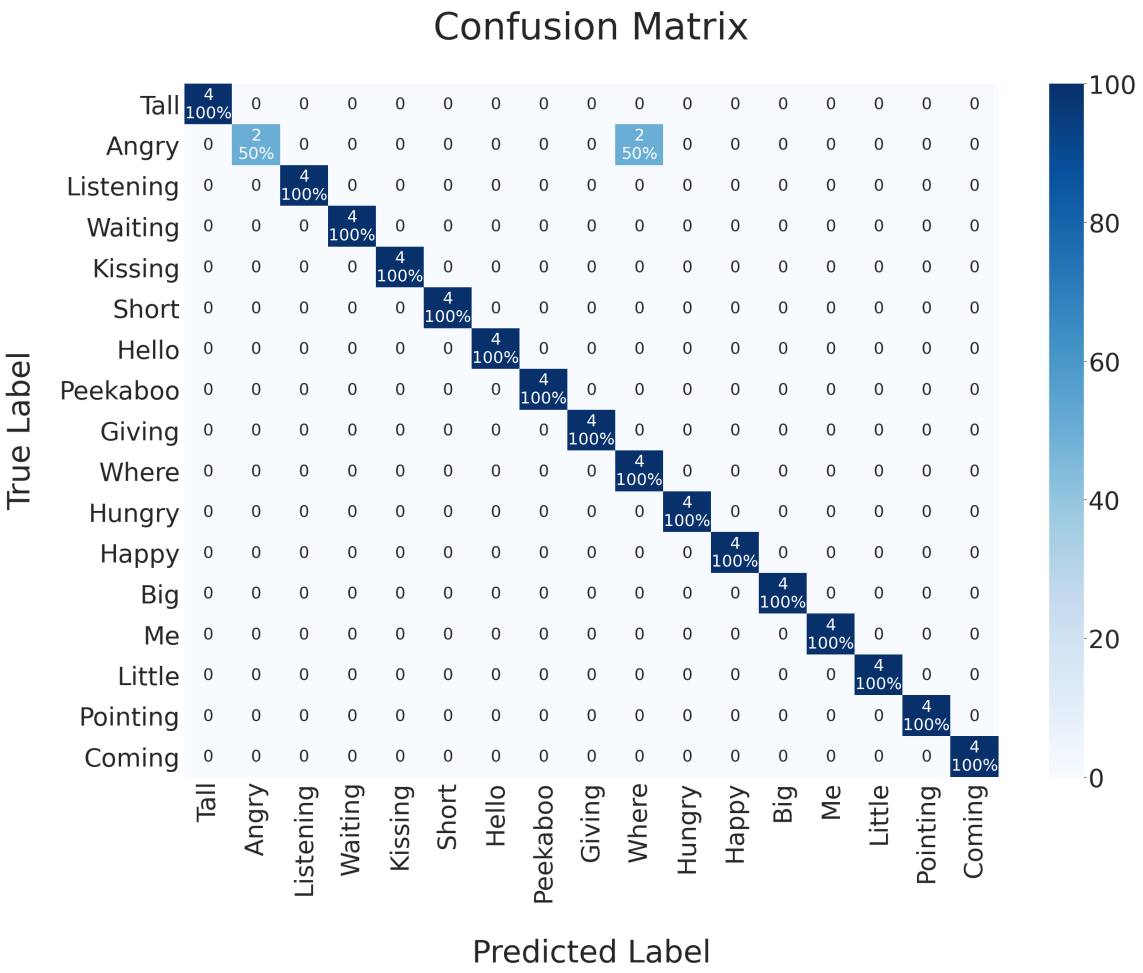

Figure 4.18: Confusion Matrix of the performances of the 19-gestures Model on two healthy adult subjects.

Table 4.20: Kinect-only acquisitions: Accuracy, F1-score, Precision and Recall for 17 gestures.

Gesture	**Accuracy (%)**	**F1(%)**	**Precision(%)**	**Recall(%)**
Tall	100	100	100	100
Angry	97	67	100	50
Listening	100	100	100	100
Waiting	100	100	100	100
Kissing	100	100	100	100
Short	100	100	100	100
Hello	100	100	100	100
Peekaboo	100	100	100	100
Giving	100	100	100	100
Where	97	80	67	100
Hungry	100	100	100	100
Happy	100	100	100	100
Big	100	100	100	100
Me	100	100	100	100
Little	100	100	100	100
Pointing	100	100	100	100
Coming	100	100	100	100

Table 4.21: Kinect-only configuration: Accuracy, F1-score, Precision and Recall for 17 gestures.

Configuration	**Accuracy (%)**	**F1(%)**	**Precision(%)**	**Recall(%)**
Kinect-only	97	97	98	97

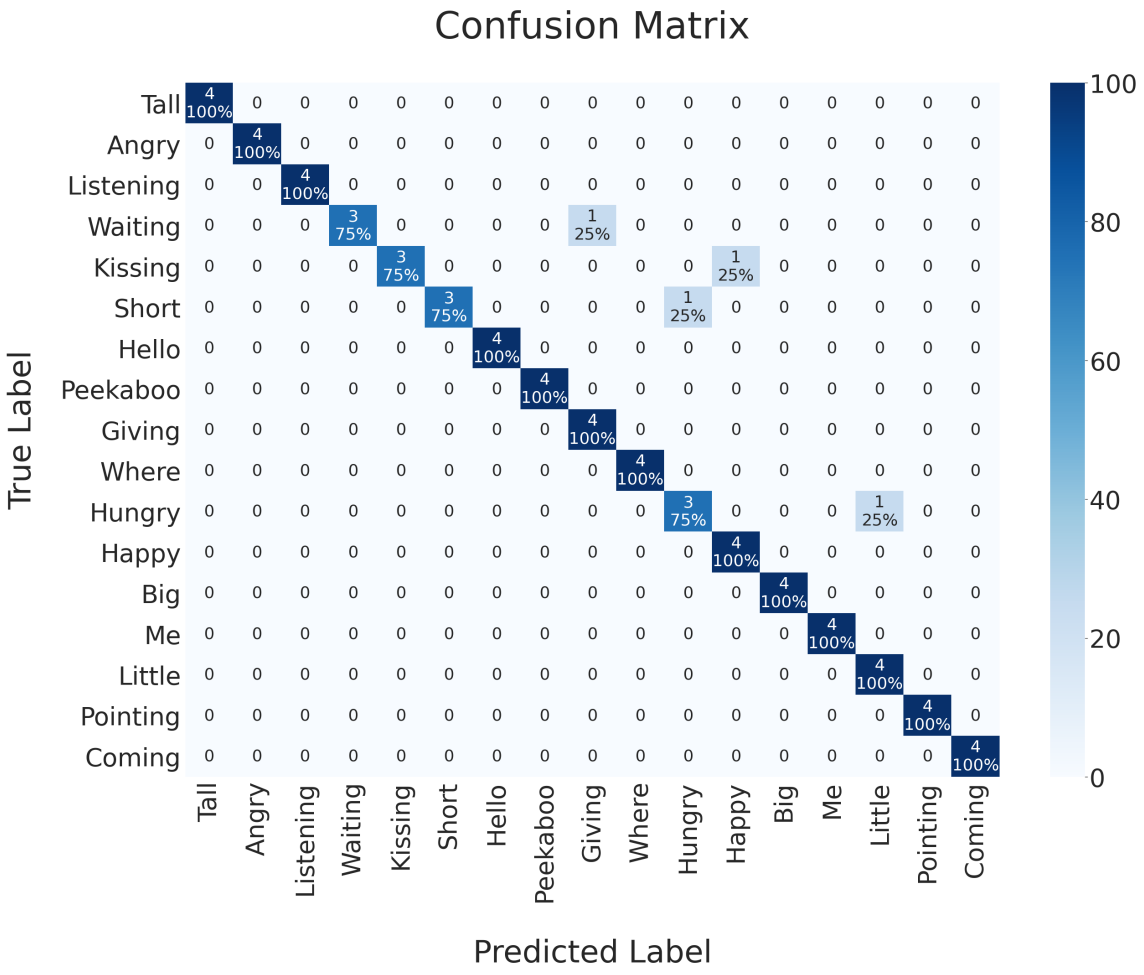

Figure 4.19: Confusion Matrix of the 19-gestures Model on two healthy adult subjects for 17 gestures

Table 4.22: Kinect-NAO acquisitions: Accuracy, F1-score, Precision and Recall for each gesture.

Gesture	Accuracy (%)	F1(%)	Precision(%)	Recall(%)
Tall	100	100	100	100
Angry	100	100	100	100
Listening	100	100	100	100
Waiting	98	86	100	75
Kissing	98	86	100	75
Short	98	86	100	75
Hello	100	100	100	100
Peekaboo	100	100	100	100
Giving	98	89	80	100
Where	100	100	100	100
Hungry	97	75	75	75
Happy	98	89	80	100
Big	100	100	100	100
Me	100	100	100	100
Little	98	89	80	100
Pointing	100	100	100	100
Coming	100	100	100	100

The other two mistaken gestures highlight how much timing is important: the preformed movements started few seconds after NAO pointed and the algorithm analyzed subject's position before the actual gesture's execution. Kinect-NAO acquisitions reached an overall accuracy of **94%** as shown in Table 4.23. These results were promising, but it has to be

Table 4.23: Kinect-NAO configuration: Accuracy, F1-score, Precision and Recall for all gestures.

Configuration	Accuracy (%)	F1(%)	Precision(%)	Recall(%)
Kinect-NAO	94	94	95	94

taken into account that the subjects were healthy adults performing gestures in a precise way.

4.5.2 @CARElab Acquisitions

At CARElab (Computer Assisted Rehabilitation) in Fondazione Don Gnocchi (Milan), new acquisitions were carried out with 6 ASD children aged between 4 and 6 part of IOGIOCO therapy. As mentioned in subsection 3.1 IOGIOCO Robot Therapy, the gesture recognition algorithm was involved only from Level 3 on. Since ASD has a wide variation in the type and severity of symptoms people can experience, children had a different way of approaching IOGIOCO therapy, thus NAO robot. For these reasons, depending on the child, different levels were reached and different gestures were performed. In Level 3, NAO's feedback was able to engage children's attention, thus increasing their interaction with the therapist. On the other hand, sometimes children lacked of interest in interacting with

NAO, thus, in these cases, it was difficult for them to keep up with therapy's exercises. During acquisitions the best models were tested.

11-gestures Model acquisitions

For the first week of acquisitions, the model able to recognize 11 gestures was experimented with the related online settings. 3 children reached Level 2 while the other 3 were able to reach Level 3 of the therapy protocol, testing the gesture recognition algorithm. Given the reduced number of samples, to analyze data in a consistent way, ASD children and therapist performances' assessments were kept together. The confusion matrix and metrics scores are shown in Figure 4.20 and in Table 4.24 respectively. Some gestures were

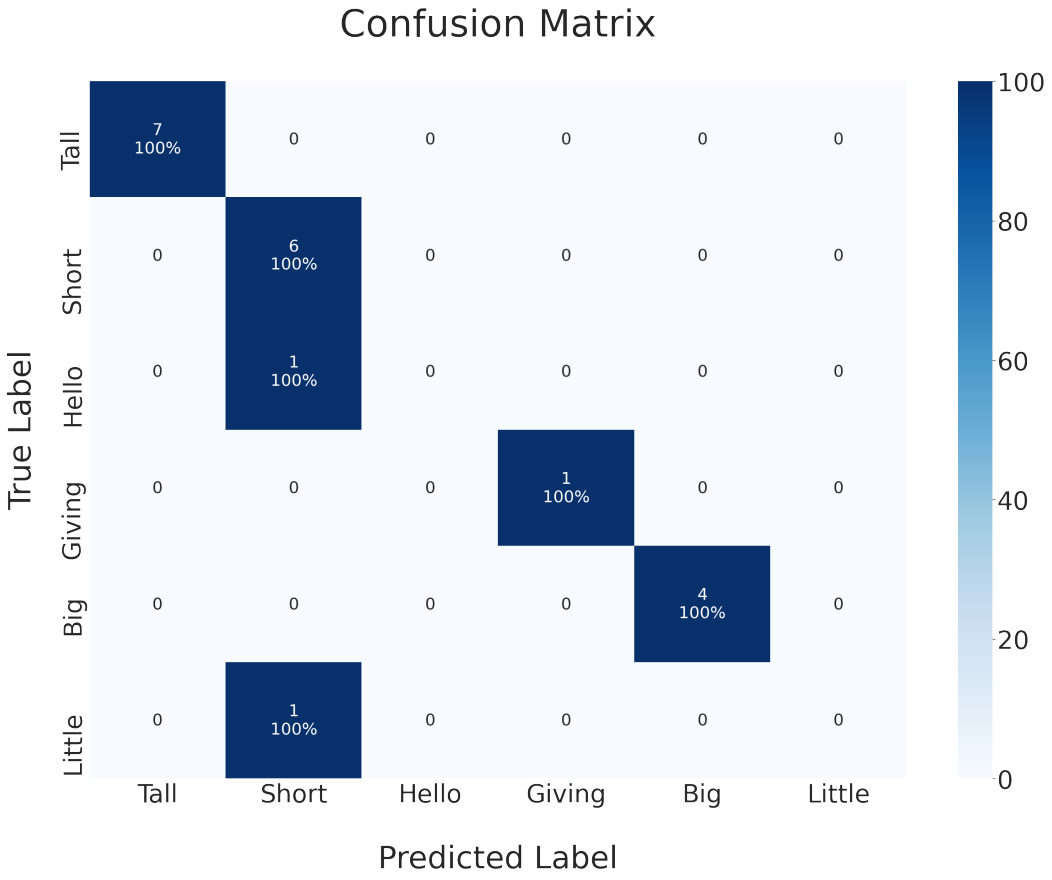

Figure 4.20: Confusion Matrix of the 11-gestures Model on three ASD children

not performed. *Hello* and *short* gestures were not correctly recognized by the algorithm. This was because actions were actually well executed, but delayed with respect to therapy protocol timing's settings. Thus, the algorithm analyzed subject's position before or after the actual gesture's execution. In Table 4.25, first week total acquisitions' metrics scores are presented: Accuracy reached **90%** while F1 score was **86%**.

19-gestures Model acquisitions

After the first week of acquisitions, the 19-gestures Model was implemented. Note that for this model, in week 2 and 3 online settings was not yet perfectly suitable as the last

Table 4.24: Metrics scores of each gesture for children's and therapist's acquisitions week 1 with 11-gestures Model

Gesture	Accuracy	F1-Score	Precision	Recall
Tall	100	100	100	100
Short	90	86	75	100
Hello	95	nan	nan	0
Giving	100	100	100	100
Big	100	100	100	100
Little	95	nan	nan	0

Table 4.25: Metrics scores of children's and therapist's acquisitions week 1 with 11-gestures Model

Week	Subjects	Accuracy (%)	F1-score (%)	Precision (%)	Recall (%)
1	Therapist and Children	90	86	82	90

version described in section 4.4.2. From week 4, the final online settings were implemented. Moreover, as the therapy progressed, 4 children were able to reach Level 3 of the protocol. The results were analysed by keeping user data separate to compare ASD children's and therapist's acquisition.

Starting with children acquisitions, as can be seen from the confusion matrix (Figure 4.21) and the metrics scores (Table 4.26), almost all actions were correctly recognized by the algorithm. For what concern *hungry* gesture, from video analysis resulted that the action was actually well executed by the child. However, the subsequent raising of the other hand while performing the action made the algorithm recognize a double handed gesture, *angry*. F1 score reached **83%** (Table 4.27). It is worth noting that a lower Recall score (82%) with respect to Precision score (89%) reduces the chances for an incorrect gesture to be recognized as a correct one, and this may be beneficial for therapy sessions. In fact, it should be pointed that the net was trained on a dataset mainly composed by healthy subjects (only 2 ASD adults out of 22 subjects), challenging the recognition task for ASD users. Moreover, a lower number of acquisitions were done and not all gestures were tested in the clinical context.

For what concerns therapist acquisitions, the confusion matrix in Figure 4.22 and the metrics scores in Table 4.28 show that the *where* gesture was mistaken with short, underlining once again the importance of timings for the algorithm to proper recognize the gesture: even if the gesture was properly executed, a delay in the performance made the algorithm focus on another movement, not concerning the therapy exercise. For the same reason, *hello* gesture was confused with *short*, *coming* and *peekaboo* and *big* gesture was predicted as a *me* one. From video analysis sometimes it happened that the therapist did not respect protocols timings to perform the gesture, because she was focused on keeping the children engaged, without loosing their attention on the protocol sequence. Thus, as can be seen from Table 4.29, the accuracy reached was **79%** with an F1-score of **81%**, slightly lower than the previous one with ASD children. This result must take into account the therapist workload during the treatment.

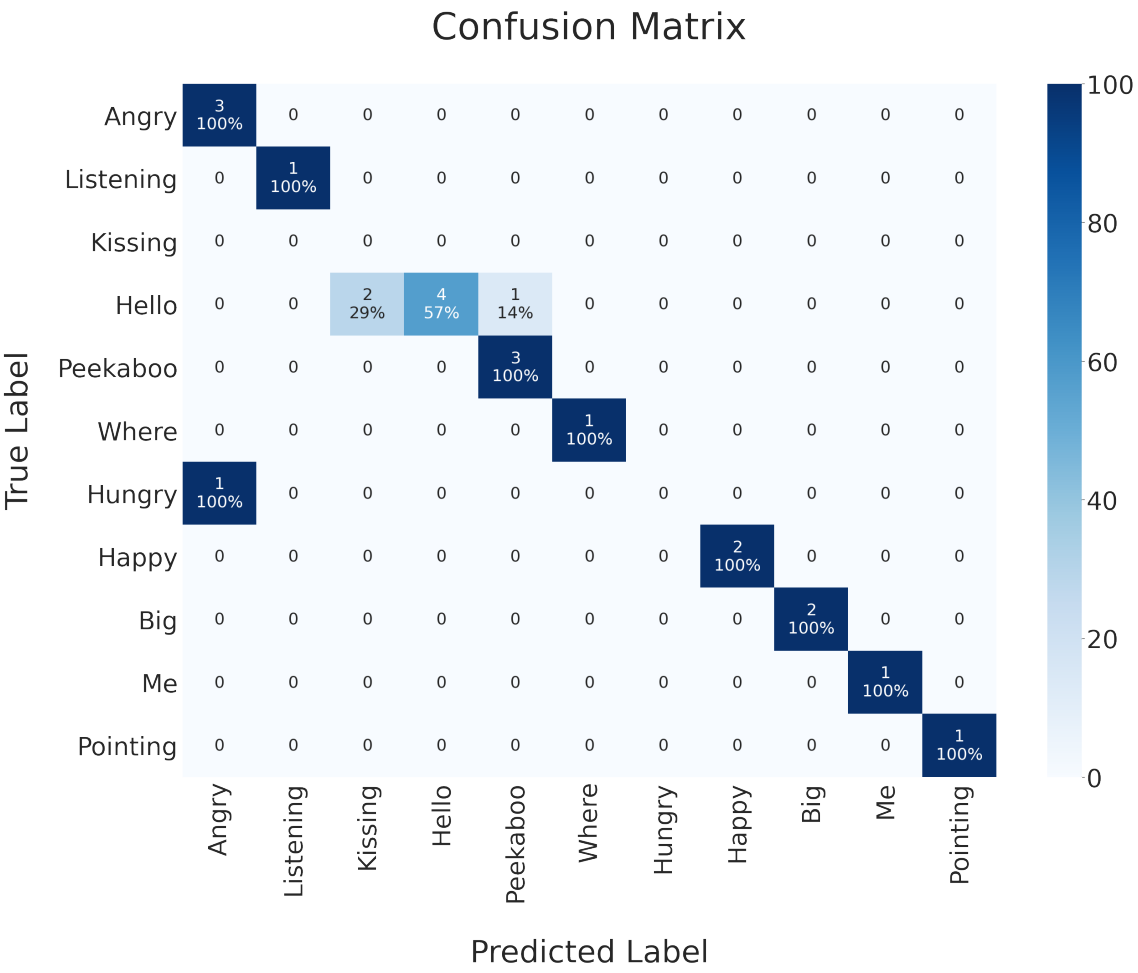

Figure 4.21: Confusion matrix of 19-gestures Model of children's performances for week 2-3-4.

Table 4.26: Metrics scores of each gesture for children's acquisitions week 2 - 3 - 4 with 19-gestures Model

Gesture	Accuracy (%)	F1-Score (%)	Precision (%)	Recall (%)
Angry	95	86	75	100
Listening	100	100	100	100
Kissing	90	nan	0	nan
Hello	86	73	100	57
Peekaboo	95	86	75	100
Where	100	100	100	100
Hungry	95	nan	nan	0
Happy	100	100	100	100
Big	100	100	100	100
Me	100	100	100	100
Pointing	100	100	100	100

Table 4.27: Metrics scores of children's acquisitions week 2 - 3 - 4 with 19-gestures Model

Week	Subjects	Accuracy (%)	F1-score (%)	Precision (%)	Recall (%)
2-3-4	Children	82	83	89	82

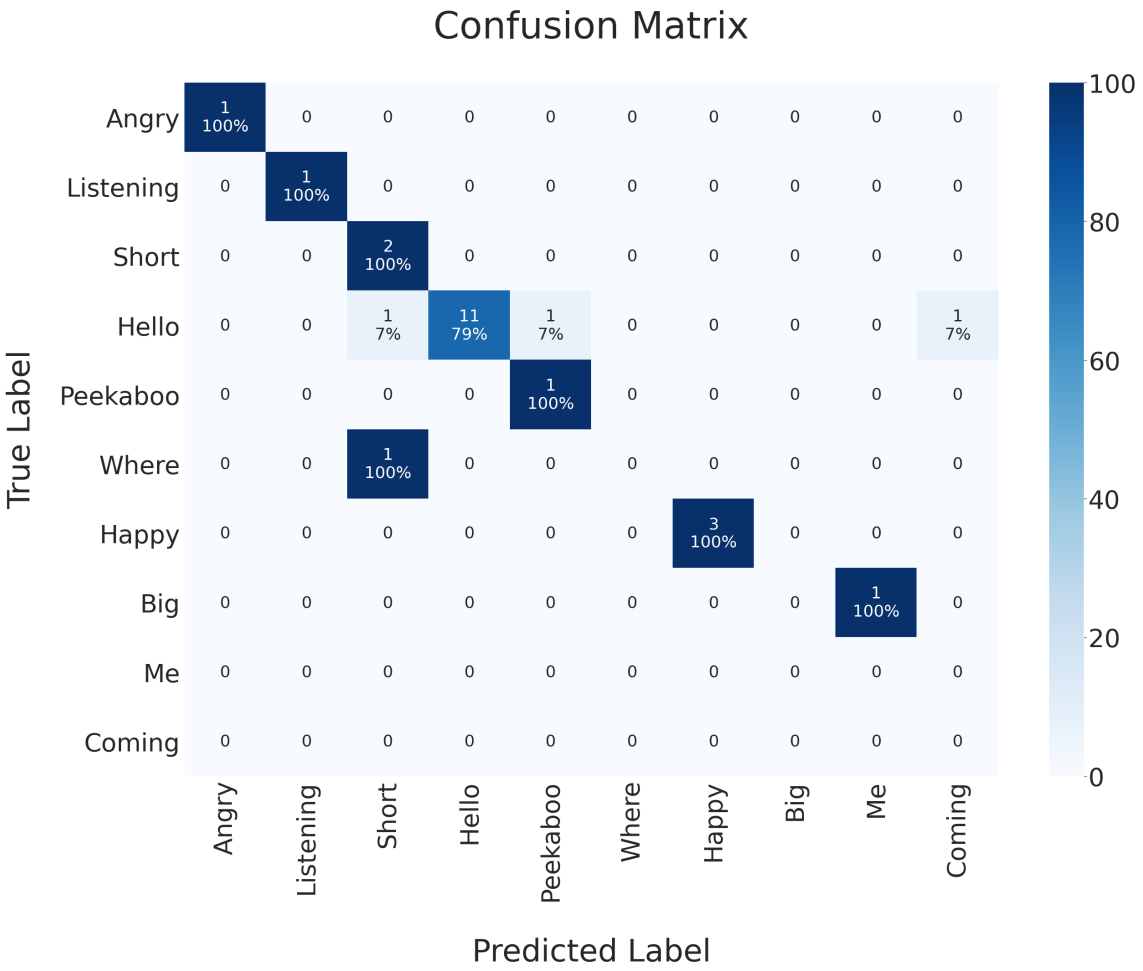

Figure 4.22: Confusion matrix of 19-gestures Model assessment of therapist's performances for week 2-3-4.

Table 4.28: Metrics scores for each gesture of therapist's acquisitions week 2 - 3 - 4 with 19-gestures Model

Gesture	**Accuracy (%)**	**F1-Score (%)**	**Precision (%)**	**Recall (%)**
Angry	100	100	100	100
Listening	100	100	100	100
Short	90	67	50	100
Hello	86	88	100	79
Peekaboo	95	67	50	100
Where	95	nan	nan	0
Happy	100	100	100	100
Big	95	nan	nan	0
Me	95	nan	0	nan
Coming	95	nan	0	nan

Table 4.29: Metrics scores of therapist's acquisitions week 2 - 3 - 4 with 19-gestures Model

Week	**Subject**	**Accuracy (%)**	**F1-score (%)**	**Precision (%)**	**Recall (%)**
2-3-4	Therapist	79	81	85	79

Chapter 5

Discussion

5.1 Data Processing

From the first sections of the results it is possible to conclude that reference system and subjects' normalization must be suitably chosen. Reference system was defined because, otherwise, the same action conducted in different positions (caused by the translation of the person or the camera) would result in a different pose feature, thus being translation variant. The coordinate system was centred in the *hip center* joint because it lies on the human body's sagittal plane. This allowed the generalization of the algorithm by the mirroring of movements performed with the right limb, thus making possible for the algorithm to recognize gestures independently of the dominant hand.

Since healthy and ASD adults and children with different physical structures were involved in this project, normalization was required to obtain user-invariance properties for the network. In general, the normalization value can not be too small: the calibration to a small and limited (in terms of movements) segment would be less reliable because of optical distortions. This project's results proved that the normalization value's choice makes the difference. It must be taken into account that a frame by frame normalization changed the length of the normalization segment, computed as Euclidean distance, from one frame to the adjacent ones: this is true especially when dealing with, for example, *yes* gesture or any gestures involving a forward bend of the torso. By the experiments carried out to find the more stable solution, trunk size's variance during the performance of an action resulted lower than the height because hip joint is more stable with respect to feet, especially when dealing with the Kinect system. For this reason, trunk size turned out to be the best normalization segment.

To summarize, this project's data processing resulted in the following properties:

- Translation invariance: The shift of the coordinate system with respect to a reference point (*hip center* joint) does not let the translations affect actions' pose features;
- User invariance: Each subject was normalized by its trunk size to make him/her scale-invariant;

As future implementation, it would be useful to add an "initialization" phase to the therapy protocol. In this phase the subject could be asked to perform a standard gesture in order

to get its effective size without any distortion due to either the camera's resolution or the subject's position in front of it.

5.2 Pose Features

Even the type of gesture set can affect data preparation. Since in this project actions involved only the upper body parts, it was convenient to feed the recognition algorithm only with those data. Moreover, as the algorithm had to face with kinematics and biomechanics data, it was important to preserve their physical meaning. Because of this, keypoints were ordered following the human skeleton structure. Due to the coordinate difference in 3D skeletons, proper normalizations were needed. Pham et al. [33, 9, 34] proposed to normalize skeleton joints by the maximum and minimum of all coordinates in the training dataset. However, as the normalization is conducted with respect to the entire training dataset, the resultant *pose features* were dataset-dependant. To tackle this problem, in this work normalization was made by maximum and minimum coordinates of each channel (x, y, z) of every skeleton sequence, which included all the movements to complete a gesture, thus being dataset independent. Overall, gesture normalizations' results underline the importance of a proper normalization when dealing with kinematics of human body's biomechanics.

The ultimate goal of data preparation was to make skeleton sequences as inputs of an Artificial Neural Network (ANN). Hence, to represent data, a fixed inputs shape was defined. As the number of keypoints was 16, a size of 48×48 was a good trade-off between the need of a good resolution and the need of preserving a reasonable spatial structure. This way, keypoints were tripled to achieve the resolution-height required. An $abc - abc - abc$ pose feature configuration turned out to be the best solution for the net to extract the information needed to recognize the whole action rather than focusing on single joint displacement variation. For what concerns the temporal dynamics, each action had to be set to a fixed width (i.e length in terms of number of frames) to be fed to an ANN. As a resampling filter was applied to each keypoint, the spatial structure's definition was preserved and a temporal interpolation was achieved: some action sequences' frames increased, others were reduced. Further developments could focus on keeping the different lengths of action (in terms of number of frames) to preserve the information related to gestures specific timings. The skeleton-based representations obtained were subjected to local contrast enhancement to further highlight the characteristics of the motion. This technique turned out to improve the capacity of the algorithm to analyze data because it emphasized the movements of each pose feature from the biomechanical point of view.

5.3 Classification

When data preparation was concluded and properly fit for this project's goal, the choice of a suitable classification method was essential. Literature analysis proved that neural networks are widely used in both online and offline gesture recognition. The net learns from data first and applies this knowledge to new one then. It is important to point out that artificial neural networks arrive at complex answers with a lack of transparency because they create rules by learning from samples, adjusting weights and evaluating measures without detailing how. For this reason, inputs characteristics and processing

were essential, especially when dealing with neural networks, since inputs data were the most reality-related part of the method's implementation. In fact, even thought the way nets learned to recognize actions can not be well understood, a proper data preparation drove the learning process. Residual learning turned out to be a good solution to extract with precision relevant features from biomechanical sequences, allowing a fast training process. Since the algorithm was developed with the aim of recognizing gestures in a real-time therapy for ASD children, the rules and patterns it learned offline from datasets affected its behaviour in the online recognition task. Speaking of that, in this project datasets included only healthy subjects (both adults and children) except for two ASD adults. It would be important that future work focuses on updating the existing dataset. The data collected from acquisitions could be used to extend the existing one with more samples of both healthy adult subjects and ASD children. Thus, the algorithm would learn to identify gestures differently performed by these users and could be tested on more subjects, becoming more generalized and robust. Another field of research could include the possible implementation of a "memory module" able to remember the features extracted by pre-trained networks (such as this work's ResNet) during the online classifications and to exploit them in the next ones to output a more solid prediction. Other networks type could be exploited and tested in sight of an online recognition. Recurrent Neural Network such as Long Short-Term Memory (LSTM) are widely used too, thanks to their capacity to model data sequences (in terms of time) so that each sample can be assumed to be dependent on the previous ones. This could lead to better results for both offline and online recognition.

Depending on the dataset used, different models were implemented and tested offline. As expected, a smaller gestures set made it easier for the network to learn how to recognize actions. The 5-gestures model reached an offline test accuracy of 94% even if data preparation was not yet the most suitable for this project's characteristics. With a larger gesture set, data preparation was essential since some movements had similar action ranges and could be easily mistaken. A stable normalization value within a skeleton sequence was also crucial to improve pose features, thus facilitating the recognition task for the ResNet, and the Data Mirroring enabled the possibility to identify also left-handed gestures. So far, an offline 95% test accuracy has been reached with the whole gesture set. This result was encouraging considering:

- the wide gesture set;
- the different temporal dynamics and duration of actions;
- the **100%** test accuracy reached by Pham et al. [33, 9, 34] with comparable methods, but dividing their gesture set into more subsets in order to implement a different model for each of them.

5.4 Online Recognition

Online recognition depends on data acquisitions instruments. In this project only Kinect camera was used, even if other wearable capturing system would have helped in a better position detecting of both therapist and child. Since IOGIOCO therapy protocol involves

children with ASD, Kinect camera was the most suitable non-intrusive option, but its behaviour made it difficult to choose online settings. In fact, since the Kinect sampling frequency decreased with robot connection, settings changed reducing both window's *size* and window's *step*. Kinect's sampling frequency behaviour and noise with robot connection should be considered in a future implementation to find a proper method to keep them apart during acquisitions or to establish a connection without reducing Kinect's ability to capture data.

In both configurations (Kinect-NAO and Kinect-only), finding the proper parameters in order to take into account the wide gesture set and the different action lengths, in terms of duration, was crucial. Some gestures, such as *kissing* or *angry* are performed in a short period of time thus, in this cases, a short window *size* would be enough. However, other gestures, such as *happy* or *listening*, would need more frames to be properly analyzed. When window *size* and *step* were not set yet, *happy* gesture was mistaken with *tall* because, if considering only the first part of *happy* action, the two movements are comparable and easily confused. Moreover, gestures can be performed with more repetitions (for instance waving multiple times or just one) and with different velocity. IOGIOCO protocol required that, within about 10 seconds, a performance assessment was established. If multiple movements were executed in that time-span, even not relevant for the treatment, the algorithm could insert those gestures into the analysis, making erroneous the evaluation. Investigating time settings by which the recognition starts might prove important in the mis-classification of gestures which were delayed performed with respect to the therapy protocol's settings. In addition, if the action was performed too slowly or too fast, or with joints' positions never seen before (so, never learned from the algorithm), the assessment could be wrong. Even in this case, a dataset updating with increased number of samples would improve the overall results.

5.5 Acquisitions

When the algorithm was properly set, new acquisitions were carried out both at Politecnico di Milano and at CARElab in Fondazione Don Gnocchi with different models. With healthy adult subjects, in the most protocol-like configuration (Kinect-NAO), the recognition capacity of the algorithm was successful and comparable with the offline results: acquisitions reached an overall accuracy of **94%**. These results were evidence of the proper online settings for the algorithm to recognize a wide range of actions part of the protocol. Only *yes* and *no* predictions were discarded because of their small and similar action ranges. The algorithm frequently identified those gestures as performed. Further developments would be needed to better understand if this behaviour was due to Kinect inaccurate caption of skeletal joint movements of small body segments or inappropriate online setting for gestures performed within few seconds. Even in this case, Kinect-NAO configuration acquisitions keeping Kinect and NAO apart (or establishing a FPS decreasing-free connection) should improve the recognition task.

3 weeks tests with the 19-gestures recognition model tested on 4 ASD children resulted in an accuracy of 83% while for therapist tests the accuracy reached was 79%. Clinical acquisitions' results confirmed the importance of inputs' information content for training an ANN. As already mentioned, only 2 out of 22 adults in Expanded Dataset had ASD.

Thus, for ASD children acquisitions, a lower accuracy was expected with respect to the other acquisitions. On the other hand, therapist acquisitions sightly lower results must be correctly interpreted. In those cases, the non-adherence to protocol timings or the misperformed gestures were due to the therapist focusing on keeping the child engaged in the learning therapy. The overall results of clinical acquisitions pointed out that a proper feedback in a teaching-gesture therapy helped the therapist in empowering the child's social skills with robot first, and subsequently with humans. In fact, the presence of robots in the therapy facilitate child's interaction with humans with respect to therapist-only treatments. Moreover, both positive and negative feedback motivated the children in improving their actions mimicking therapist's gestures and also in repeating the exercises. Customized stimuli may be more effective at eliciting skills learning especially when dealing with ASD children. In point of that, child-specific feedback might improve the effectiveness of the therapy. Furthermore, the implementation should also consider the next challenging protocol's levels, in which gesture teaching is inserted in a story-telling scenario.

Chapter 6

Conclusions and Future Work

The cause of ASD is still being studied and its complete understanding has not been reached yet. Many facets characterize ASD and it is likely caused by many factors. For these reasons, a single cure or solution has not been found so far. Newest treatments explore motor therapies to help ASD children social-interacting in their every-day life. This thesis' work was integrated within IOGIOCO therapy, an interactive mirroring robotics game which aims at helping children in developing imitation, motor and gesture skills. The project demonstrated successfully the use of a gesture recognition algorithm for the purpose of increasing ASD children engagement and empowering gestures' learning by means of a straightforward and robust feedback system in a triadic interaction between therapist, robot and child.

In this project, skeleton data were used to interpret gestures, as human actions can be described by the movements of skeleton joints. Data extracted from a Kinect camera were properly processed to control and drive net's learning as much as possible. An offline version of the classification algorithm was first designed in sight of an online implementation. A ResNet was exploited to generate the offline model for its ability of providing deeper neural networks able to extract more and more features. The offline recognition model was able to analyze 19 types of gestures and reached an accuracy of **95%**. These results were promising considering the wide gesture set, the different temporal dynamics and duration of actions. A Kinect camera was used to track the human body in real-time. A recognition system able to classify both therapist and children's actions in a clinical context was set up. The online recognition algorithm was tested on 2 healthy adult subjects, obtaining an accuracy of **97%**, and on 4 ASD children, reaching an accuracy of **83%**. These results were encouraging since the net was trained on a dataset mainly composed by healthy subjects (only 2 ASD adults out of 22 subjects), thus challenging the recognition task for ASD users. Moreover, it has to be taken into account that a lower number of acquisitions were done and not all gestures were tested in the clinical context. Children had different ways of approaching IOGIOCO therapy, thus different engagement levels were detected. Even though not all children managed to keep up with the therapy, robot's feedback were able to increase their attention's level.

It would be important that future work focuses on updating the training dataset which the model was built on. In particular, the data collected from acquisitions could be used to extend the existing one with more samples of both healthy adult subjects and ASD

children. In this way, the algorithm would learn to identify gestures differently performed by these users and could be tested on more subjects, becoming more generalized and robust. Another field of research could include the possible implementation of a "memory module" able to remember the features extracted by pre-trained networks (such as this work's ResNet) during the online classifications and to exploit them in the next ones to output a more solid prediction. Moreover, Kinect's FPS behaviour and noise with robot connection should be considered in a future implementation to find a proper method to keep them apart during acquisitions or to establish a connection without lowering Kinect's ability of capturing information. Investigating time settings by which the recognition starts might prove important in the mis-classification of gestures which were delayed performed with respect to the therapy protocol's settings. This is an issue for future research to explore. Also it would be useful to add an "initialization" phase to the therapy protocol. In this phase the subject could be asked to perform a standard gesture in order to get its effective size without any distortion due to either the camera's resolution or the subject's position in front of it. Another important aspect is the type of feedback the robot provides. When dealing with such a broad range of conditions as in the ASD, customized stimuli may be more effective at eliciting skills learning. In point of that, child-specific feedback might improve the effectiveness of the therapy. Furthermore, the next challenging protocol's levels should be considered in the implementation in which gesture teaching and recognition system would be integrated in a story-telling scenario.

Up to date, children involved in the robotics therapy are few and there is no unequivocal evidence that observed improvements will last. A Randomized Controlled Trial (RCT) would reduce biases when testing this treatment's efficacy. However, the overall clinical outcomes were encouraging, demonstrating a successful integration of Biomedical Engineering in therapeutic applications.

Bibliography

[1] A. Y. Onaolapo and O. J. Onaolapo, "Global data on autism spectrum disorders prevalence: A review of facts, fallacies and limitations," *Universal Journal of Clinical Medicine*, vol. 5, no. 2, pp. 14–23, 2017.

[2] C. Duarte, L. Carriço, D. Costa, D. Costa, A. Falcão, and L. Tavares, "Welcoming gesture recognition into autism therapy," *Conference on Human Factors in Computing Systems - Proceedings*, 04 2014.

[3] E. Marinoiu, M. Zanfir, and C. Sminchisescu, "3d human sensing, action and emotion recognition in robot assisted therapy of children with autism," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2158–2167, 2018.

[4] H. Cao, P. G. Esteban, M. Bartlett, P. Baxter, T. Belpaeme, E. Billing, H. Cai, M. Coeckelbergh, C. Costescu, D. David, A. De Beir, D. Hernandez, J. Kennedy, H. Liu, S. Matu, A. Mazel, A. Pandey, K. Richardson, E. Senft, S. Thill, G. Van de Perre, B. Vanderborght, D. Vernon, K. Wakanuma, H. Yu, X. Zhou, and T. Ziemke, "Robot-enhanced therapy: Development and validation of supervised autonomous robotic system for autism spectrum disorders therapy," *IEEE Robotics Automation Magazine*, vol. 26, no. 2, pp. 49–58, 2019.

[5] S. Sial, "Robot assisted therapy for children with autism spectrum disorder - a survey," *Robotics and Automation Engineering Journal*, vol. 2, 03 2017.

[6] M. Licari, G. Alvares, K. J. Varcin, K. L. Evans, D. Cleary, S. Reid, E. Glasson, K. Bebbington, J. E. Reynolds, J. Wray, and A. Whitehouse, "Prevalence of motor difficulties in autism spectrum disorder: Analysis of a population-based cohort," *Autism Research*, vol. 13, 2019.

[7] D. McAuliffe, Y. Zhao, A. S. Pillai, K. Ament, J. H. Adamek, B. Caffo, S. Mostofsky, and J. B. Ewen, "Learning of skilled movements via imitation in asd," *Autism Research*, vol. 13, 2019.

[8] I. Riquelme, S. Hatem, and P. Montoya, "Abnormal pressure pain, touch sensitivity, proprioception, and manual dexterity in children with autism spectrum disorders," *Neural Plasticity*, vol. 2016, 2016.

[9] H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. Velastin, "Learning and recognizing human action from skeleton movement with deep residual neural networks," *ArXiv*, vol. abs/1803.07780, 2018.

[10] M. Asadi-Aghbolaghi, A. Clapés, M. Bellantonio, H. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera, "A survey on deep learning based approaches for action and gesture recognition in image sequences," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 476–483, 2017.

[11] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton optical spectra-based action recognition using convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 807–811, 2018.

[12] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[13] L. Yu, S. Wang, and K. K. Lai, "Data preparation in neural network data analysis," 2007.

[14] D. Bai, B. Yip, G. Windham, A. Sourander, R. Francis, R. Yoffe, E. Glasson, B. Mahjani, A. Suominen, H. Leonard, M. Gissler, J. Buxbaum, K. Wong, D. Schendel, A. Kodesh, M. Breshnahan, S. Levine, E. Parner, S. N. Hansen, C. Hultman, A. Reichenberg, and S. Sandin, "Association of genetic and environmental factors with autism in a 5-country cohort.," *JAMA psychiatry*, 2019.

[15] S. J. Rogers, I. Cook, and A. Meryl, *Handbook of autism and pervasive developmental disorders: Diagnosis, development, neurobiology, and behavior*, ch. Imitation and Play in Autism., pp. 382–405. F. R. Volkmar, R. Paul, A. Klin, and D. Cohen, 2005.

[16] S. J. Rogers, S. L. Hepburn, T. Stackhouse, and E. Wehner, "Imitation performance in toddlers with autism and those with other developmental disorders.," *Journal of Child Psychology and Psychiatry*, vol. 44, no. 5, pp. 763–781, 2003.

[17] I. M. Smith, "Gesture imitation in autism i: Nonsymbolic postures and sequences.," *Cognitive Neuropsychology*, vol. 15, no. 6-8, pp. 747–770, 1998.

[18] G. Huguet, M. Benabou, and T. Bourgeron, "The genetics of autism spectrum disorders.," *Part of the Research and Perspectives in Endocrine Interactions book series*, pp. 102–103, 2016.

[19] A. Knopf, "Autism prevalence increases from 1 in 60 to 1 in 54: Cdc," *The Brown University Child and Adolescent Behavior Letter*, vol. 36, pp. 4–4, 2020.

[20] A. Masi, M. M. DeMayo, N. Glozier, and A. J. Guastella, "An overview of autism spectrum disorder, heterogeneity and treatment options.," *Neuroscience Bulletin*, pp. 183–193, 2017.

[21] S. R. Sharma, X. Gonda, and F. I. Tarazi, "Autism spectrum disorder: classification, diagnosis and therapy," *Pharmacology & therapeutics*, vol. 190, pp. 91–104, 2018.

[22] R. Loomes, L. Hull, W. Polmear, and M. Locke, "What is the male-to-female ratio in autism spectrum disorder? a systematic review and meta-analysis.," *Journal of the American Academy of Child & Adolescent Psychiatry*, pp. 466–474, 2017.

[23] T. Bourgeron, "From the genetic architecture to synaptic plasticity in autism spectrum disorder.," *Nature Reviews Neuroscience*, vol. 16, pp. 551–563, 2015.

[24] R. Bernier, A. Mao, and J. Yen, "Psychopathology,families, and culture:autism," *Child and Adolescent Psychiatric Clinics of North America*, vol. 19, pp. 855–867, 2010.

[25] M. Coeckelbergh, C. Pop, R. Simut, A. Peca, S. Pintea, D. David, and B. Vanderborght, "A survey of expectations about the role of robots in robot-assisted therapy for children with asd: Ethical acceptability, trust, sociability, appearance, and attachment," *Science and Engineering Ethics*, vol. 22, pp. 47–65, 2016.

[26] B. Scassellati, H. Admoni, and M. Matarić, "Robots for use in autism research.," *Annual review of biomedical engineering*, vol. 14, pp. 275–94, 2012.

[27] P. Esteban, P. Baxter, T. Belpaeme, E. Billing, H. Cai, H.-L. Cao, M. Coeckelbergh, C. A. Costescu, D. David, A. Beir, Y. Fang, Z. Ju, J. Kennedy, H. Liu, A. Mazel, A. Pandey, K. Richardson, E. Senft, S. Thill, G. Perre, B. Vanderborght, D. Vernon, H. Yu, and T. Ziemke, "How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder," *Paladyn, Journal of Behavioral Robotics*, vol. 8, pp. 18 – 38, 2017.

[28] A. Zaraki, L. Wood, B. Robins, and K. Dautenhahn, "Development of a semi-autonomous robotic system to assist children with autism in developing visual perspective taking skills," *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 969–976, 2018.

[29] Z. Zheng, E. M. Young, A. Swanson, A. S. Weitlauf, Z. Warren, and N. Sarkar, "Robot-mediated mixed gesture imitation skill training for young children with asd," *2015 International Conference on Advanced Robotics (ICAR)*, pp. 72–77, 2015.

[30] S. Attawibulkul, N. Sornsuwonrangsee, W. Jutharee, and B. Kaewkamnerdpong, "Using storytelling robot for supporting autistic children in theory of mind," *International Journal of Bioscience, Biochemistry and Bioinformatics*, vol. 9, pp. 100–108, 2019.

[31] F. Giuliani, B. C. Marchetti, V. Perrenoud, and P. E. Korh, "Is storytelling therapy useful for children with autism spectrum disorders and severe mental retardation," *Advanced techniques in biology and medicine*, vol. 4, pp. 1–5, 2016.

[32] A. Papadakis, E. Mathe, I. Vernikos, A. Maniatis, E. Spyrou, and P. Mylonas, "Recognizing human actions using 3d skeletal information and cnns," in *EANN*, 2019.

[33] H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. Velastin, "Exploiting deep residual networks for human action recognition from skeletal data," *ArXiv*, vol. abs/1803.07781, 2018.

[34] H.-H. Pham, H. Salmane, L. Khoudour, A. Crouzil, P. Zegers, and S. Velastin, "Spatio-temporal image representation of 3d skeletal movements for view-invariant action recognition with deep convolutional neural networks," *Sensors*, vol. 19, 2019.

[35] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 579–583, 2015.

[36] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "Rgb-d-based human motion recognition with deep learning: A survey," *ArXiv*, vol. abs/1711.08362, 2018.

[37] E. Mathe, A. Mitsou, E. Spyrou, and P. Mylonas, "Arm gesture recognition using a convolutional neural network," in *2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pp. 37–42, IEEE, 2018.

[38] L. Zhang, G. Zhu, P. Shen, J. Song, S. Afaq Shah, and M. Bennamoun, "Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 3120–3128, 2017.

[39] Y. Zhu and C. Huang, "An adaptive histogram equalization algorithm on the image gray level mapping," *Physics Procedia*, vol. 25, pp. 601–608, 2012.

[40] A. Lee, P. Taylor, J. Kalpathy-Cramer, and A. Tufail, "Machine learning has arrived!," *Ophthalmology*, vol. 124, no. 12, pp. 1726–1728, 2017.

[41] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using kinect," *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 185–188, 2012.

[42] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 188–195, 2010.

[43] A. Anuj, T. Mallick, P. Das, and A. Majumdar, "Robust control of applications by hand-gestures," *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pp. 1–4, 2015.

[44] Y. Gu, H. Do, Y. Ou, and W. Sheng, "Human gesture recognition through a kinect sensor," *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1379–1384, 2012.

[45] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden markov model structure for information extraction," in *Proceedings of AAAI'99 Workshop on Machine Learning for Information Extraction*, January 1999.

[46] S. Bhattacharya, B. Czejdo, and N. Pérez, "Gesture classification with machine learning using kinect sensor data," *2012 Third International Conference on Emerging Applications of Information Technology*, pp. 348–351, 2012.

[47] E. Gani and A. Kika, "Albanian sign language (albsl) number recognition from both hand's gestures acquired by kinect sensors," *ArXiv*, vol. abs/1608.02991, 2016.

[48] D. A. Maharani, H. Fakhrurroja, Riyanto, and C. Machbub, "Hand gesture recognition using k-means clustering and support vector machine," in *2018 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, pp. 1–6, 2018.

[49] Z. Zeng, Q. Gong, and J. Zhang, "Cnn model design of gesture recognition based on tensorflow framework," *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1062–1067, 2019.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[51] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[52] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3633–3642, 2017.

[53] G. Luzhnica, J. Simon, E. Lex, and V. Pammer-Schindler, "A sliding window approach to natural hand gesture recognition using a custom data glove," *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 81–90, 2016.

[54] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4207–4215, 2016.

[55] F. B. Baldissera and F. Vargas, "A light implementation of a 3d convolutional network for online gesture recognition," *IEEE Latin America Transactions*, vol. 18, pp. 319–326, 2019.

[56] E. Mota, A. Moreira, and T. Nascimento, "Motion and teaching of a nao robot," 2011.

[57] S.-W. Sun, T.-C. Mou, C.-C. Fang, P. Chang, K. Hua, and H. Shih, "Baseball player behavior classification system using long short-term memory with multimodal features," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[58] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019.