

POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering



Master's Thesis

Creation of Hand Motion Capture Dataset to Generate AI-based Task Classifiers to Improve Human-Robot Collaboration in Industry 4.0

Supervisors

Prof. Danilo DEMARCHI

Ph.D. Matteo MENOLOTTO

Ph.D. Brendan O'FLYNN

Candidate

Francesca MONGELLI

Academic Year 2022/2023
March 2023

Acknowledgement

Per questo lavoro di tesi vorrei ringraziare prima di tutto il mio relatore il **prof.sor Danilo Demarchi**, senza il quale non avrei potuto vivere quest'esperienza all'estero. Insieme a lui vorrei ringraziare il dottor **Menolotto** e il dottor **O'Flynn** per avermi seguito e supportato durante i sei mesi in Irlanda.

Alla mia **famiglia**, che mi ha supportato in tutte le scelte senza mai farmelo pesare, grazie perché senza di voi nulla di tutto questo sarebbe stato possibile. Ad **Anna**, grazie per ogni parola spesa per supportarmi o darmi la forza di andare avanti anche nei momenti più difficili, non è scontato e sei stata fondamentale.

Ai miei amici, a quelli che hanno vissuto le pene e le gioie di questa università giorno dopo giorno al mio fianco: **Cosimo, Noemi e Susanna**. Inutile dirvi che questo percorso non sarebbe stato lo stesso senza di voi, Cosimo sei stato la gioia del lunedì mattina e le nuove canzoni del venerdì, Memi sei stata la spalla nei momenti no e nei momenti spritz, Susi sei stata la parte razionale che mancava al quartetto.

Agli amici che hanno preso parte in questo percorso in grande o piccola parte: **Dario, Silvia, Sardo, Gian, Frank, Lale, Vale** e tanti altri che se dovessi menzionare nome per nome non basterebbe questa tesi. Grazie perché ognuno di voi ha contribuito a costruire la persona che sono oggi.

Agli amici conosciuti in Irlanda, gli "**Andrea, Luca, Alida, Eoghan, Ricardo, Alessandro e Semira**", avete contribuito a rendere quest'esperienza un ricordo indimenticabile.

Table of Contents

Table of Contents.....	2
Table of Figures	4
Table of Tables.....	6
Abbreviations	7
Abstract.....	8
1 Introduction	10
2 State of Art.....	12
2.1 Hand	12
2.1.1 Hand's Anatomy	12
2.1.2 Biomechanical models for the hand	13
2.2 Motion Capture.....	18
2.2.1 Camera based system.	19
2.2.2 Non camera-based systems	24
2.2.3 Features of each technique	29
2.3 Data Gloves	30
2.3.1 Glove types.....	31
2.3.2 Capability	32
2.3.3 Wearability	32
2.4 Open Access Database	33
2.5 AI classifier	35
2.5.1 Typologies of SL.....	37
3 Materials.....	40
3.1 Camera-based Equipment	40
3.2 Smart Glove Equipment	41
3.3 Lab setup	42
3.4 3D printer objects and Power point instructions developed	44
4 Method	46
4.1 Database Creation.....	46
4.1.1 Data acquisition	46
4.1.2 Tasks explanation	47
4.1.3 Data processing.....	50
4.2 Features identification	52
4.2.1 Features Database creation	54

Table of Contents

4.3	AI classification.....	54
4.3.1	Approach A: Basic Accuracy.....	54
4.3.2	Approach B: Manual k-fold cross-validation.....	56
4.3.3	Approach C: Balanced Dataset.....	58
5	Results	60
5.1	Database	60
5.2	AI classification.....	62
5.2.1	Approach A: Basic Accuracy.....	62
5.2.2	Approach B: Manual k-fold cross-validation.....	63
5.2.3	Approach C: Balanced Dataset.....	65
6	Discussion	69
6.1	HANDMI4 Dataset.....	69
6.2	AI Classifier.....	71
6.2.1	Approach A.....	72
6.2.2	Approach B.....	73
6.2.3	Approach C.....	74
6.2.4	Summary analysis of Results.....	75
7	Conclusions and future works	77
	References.....	80
	Appendix A	84
	Appendix B	87

Table of Figures

Figure 1. Representation of human hand with its joint and bones [10].....	13
Figure 2. Circular (on the left) and prismatic (on the right) grasping reconstruction. a) Original Gesture b) Hand model reconstructed by 9 elements c) Hand Model reconstructed by 6 elements [8].....	15
Figure 3. Representation of Morecki's model [11]	16
Figure 4. 15 DoF model's with the description of the DoF used [5]	17
Figure 5. A 21 DoF's model [7]	18
Figure 6. Optical motion capture process	20
Figure 7. The principle of camera triangulation.[4]	20
Figure 8. Different size of passive markers [3]	22
Figure 9. An example of active markers [2]	23
Figure 10. An example of marker less MoCap [1].....	23
Figure 11. Electromechanical systems. On the left an example of optical fibre while on the right a potentiometer one [6].....	24
Figure 12. An example of magnetic tracking system [9].....	26
Figure 13. The Tyndall data glove, with IMU inside, use in the data acquisition of this study.....	26
Figure 14. An example of MEMS sensor. [12].....	27
Figure 15. Pictures some examples of smart gloves. [13]	31
Figure 16. AI taxonomy	37
Figure 17. PrimeX13 camera on the left and PrimeX41 on the right.....	40
Figure 18. Placement of markers on the hand	41
Figure 19. Tyndall Smart glove.....	42
Figure 21. Aluminium cage for initial cameras disposition.....	43
Figure 20. Final lab setup for the study, in the red cube the shooting volume (1mx1m)	43
Figure 22. Projects in Solidworks (above) and printed objects (below)	44
Figure 23. PowerPoint instructions created for the process. (a) example of Taxonomy's slide, (b) example of Industry's slide, (c) overall picture of the presentation.	45
Figure 24. Workflow of the data acquisition process	47
Figure 25. Cutkosky Grasp Taxonomy Task	48
Figure 26. Industry tasks. (A) The two boxes used for pick-and-place. (B) The cubic object to observe during object inspection (C) A seven-piece puzzle to be assembled during the hand assembly task. (D) Assembly components and tools for the last task.....	49
Figure 27. Motive Software interface. (A) Builder, where to create the hand model. (B) Gaps where to fill the gap. (C) Curves where to do the smoothing. (D) Labels where adjust the labels. (E) Hand model made with the Builder	51
Figure 28. A scheme of 10-fold cross validation applied in Approach A, where the grey box is for training and the yellow one for validation.	55
Figure 29. Manual 3-fold cross validation. The subjects framed in red are the left-handers. While the ochre rectangle is the validation fold and the grey the training ones.	57
Figure 30. The 15 top features choose for Glove classification.	58
Figure 31. The 15 top features choose for Camera classification.	59
Figure 32. HANDMI4 partition	60

Table of Figures

Figure 33. Confusion matrices for the chosen network in the camera dataset	63
Figure 34. Confusion matrices for the chosen network in the glove dataset	64
Figure 35. Confusion matrix of validation and test for the Neural Network net.....	66
Figure 36. Confusion matrix of validation and test for the Random Forest net.....	66
Figure 37. Confusion matrix of validation and test for the SVM net.....	66
Figure 38. Confusion matrix of validation and test for the SVM net.....	67
Figure 39. Confusion matrix of validation and test for the Neural Network net.....	67
Figure 40. Confusion matrix of validation and test for the Random Forest net.....	67

Table of Tables

Table 1. Pros and Cons of the systems analysed above.	29
Table 2. Overview of available open access dataset in the context of hand gesture.	35
Table 3. Camera-based specs' system [56].	40
Table 4. Demographics information of participants.	46
Table 5. Statistic features extract for each file.	52
Table 6. Morphological feature calculated for each subject.	53
Table 7. File's labelling code	60
Table 8. Labelling code for .csv file in GLOVE, where Q = X, Y, Z; F= Index, Middle, Ring, Little, Thumb and ## equal to AA (Abduction/Adduction) or FE (Flexion/Extension).	61
Table 9. Labelling code for .csv file in CAMERA, where # = 0, 1, 2, 3 (in the notation a quaternion q is defined as follow: $q = q_1 i + q_2 j + q_3 k + q_0$), V = X, Y, Z and F= Palm, Thumb, Index, Middle, Ring, Little, Forearm.	62
Table 10. Classification results for camera dataset.	62
Table 11. Classification results for glove dataset.	62
Table 12. Classification results for camera dataset without feature selection.	63
Table 13. Classification results for camera dataset with 18/173 features selected with CHI2.	64
Table 14. Classification results for camera dataset with 18/173 features selected with ANOVA.	64
Table 15. Classification results for glove dataset without feature selection.	65
Table 16. Classification results for glove dataset with 18/173 features selected with CHI2.	65
Table 17. Classification results for glove dataset with 18/173 features selected with ANOVA.	65
Table 18. Classification results for glove dataset.	65
Table 19. Classification results for camera dataset.	67
Table 20. The table shows the overall accuracy [%] of the different approach. Specifically, B1 refers to the second approach without feature selection, B2 18 features selected with CHI2, and B3 18 features selected with ANOVA.	68

Abbreviations

HRC	Human-Robot Collaboration
MoCap	Motion Capture
HANDMI4	Hand Motion capture data for industry 4.0
IMU	Inertial Measurement Unit
DoFs	Degrees of Freedom
RoM	Range of Motion
DIP	Distal interphalangeal joint
PIP	Proximal Interphalangeal joint
MCP	Metacarpophalangeal joint
IP	Interphalangeal joint
CMC	Carpometacarpal joint
TM	Trapeziometacarpal joint
PCs	Principal components
FOV	Field of view
MEMS	Micro-Electromechanical Systems
HAR	Human Activity Recognition
PLA	Polylactic acid
IQR	Interquartile range
SF	Shape factor
CHI2	Chi-square test
ANOVA	Analysis of Variance
NaN	Not a Number
SVM	Support Vector Machine
RF	Random Forest
NN	Neural Network

Abstract

The Fourth Industrial Revolution, also known as Industry 4.0, is rapidly transforming industrial manufacturing processes by introducing advanced automation, intelligent edge technology and data driven AI tools to the workflow. In this context, collaborative robotics is emerging as an attractive technology to optimise production efficiencies. They exploit advanced actuation and sensing technologies to enable collaborative robots work safely in close proximity with their human co-workers. Such flexibility is allowing even middle and small-scale enterprises to implement automation despite the typically reduced footprint and layout of their small-scale manufacturing lines. This has led to an increase in demand for human-robot collaboration (HRC) technology which is able to promote effective and safe interaction between humans and robots. One of the approaches involves the collection of human-centric data to extract information which is useful in improving robot self-awareness as well as the actions being performed by the operator carrying out the task. Motion capture (MoCap) technology, and in particular hand motion data, can be used to extract task related information, although the lack of available datasets is impeding advancement in this field. Research questions which remain unanswered include the identification of MoCap technology which is well suited to tracking hand motions in constrained environments (industry), and also the definition of the optimum (minimum) number of sensors able to reliably classify tasks. This thesis aims to address some of these open questions by collecting a comprehensive hand MoCap database for task classification in the context of HRC for Industry 4.0 and investigating the performance of some AI-based classification strategies for hand motion analytics.

The open-access database developed, HANDMI4 (hand motion capture data for industry 4.0), includes hand grasp configurations and industry-relevant dynamic movement acquired with two different MoCap technology: camera-based and data glove. From this dataset a set of statistical and costume features were evaluated and labelled according to the task being carried out (four different tasks are

Abstract

investigated). The features were then used to train different artificial intelligence (AI) networks, including support vector machine, random forest, and neural network strategies.

The results obtained show that the neural network-based classifier outperformed the other AI networks for the camera-based data achieving an accuracy rate of 95%, while for the data glove a result of 95.7% accuracy was obtained using the random forest approach. Another significant result is the identification of the palm and the middle finger as the most significant anatomical segments for the task classification in terms of hand motion artefacts in an industrial context.

The availability of the HANDMI4 database as an open access tool will foster further research and development in the field of hand motion recognition for HRC. This research has significant implications for the development of HRC technologies in smart manufacturing, as it represents an essential step towards creating a safe and efficient workspace where humans and robots can collaborate seamlessly.

In conclusion, this thesis provides a comprehensive solution for hand motion recognition in the context of HRC in industry 4.0. The results demonstrate the potential impact of hand motion recognition in HRC and the usage of neural network-based AI classifiers to achieve high accuracy rates. The ultimate goal is to develop intelligent systems that can promote safe and efficient human-robot collaboration.

1 Introduction

Human-Robot Collaboration (HRC) is becoming an increasingly important aspect of the manufacturing landscape, especially with the advent of the Fourth Industrial Revolution, also known as Industry 4.0. This new era of manufacturing is characterized by the integration of advanced technologies such as artificial intelligence and robotics, which offer a new level of flexibility, efficiency, and safety. To fully realize the potential of HRC, it is essential to provide the robot with some form of autonomy. Especially in the context of the new collaborative robotics, where robots and human co-workers share the same working volume, have accurate and reliable methods to recognize and interpret human motions is paramount.

In particular, hand motion recognition is a crucial component of HRC, as it enables robots to interpret and respond to the actions of their human collaborators. There are various technics to capture and recognise human hand, including camera based MoCap and data glove technology. Camera-based MoCap involves tracking markers (usually via triangulation) placed on specific landmarks of the hand and reconstruct the movement of each marker within the global coordinate of the cameras. While smart gloves use wearable sensors (e.g. accelerometers or flexible resistors) to track the orientation (or the joint's angle) of the hand's components. Both techniques have their own strengths and weaknesses, and each has the potential to provide valuable information for motion recognition in HRC.

The aim of this thesis is to create an open-access database for Industry 4.0, using both technologies, which will be used to train an AI network for task recognition providing a comprehensive solution for motion recognition in HRC and a thorough depiction of human motions in a variety of industrial tasks.

At this end inertial measurement unit (IMU) based data glove (Tyndall smart glove), capable of tracking the hand pose via 12 9-axis IMUs, and a camera-based MoCap system equipped with 12 cameras from Optitrack (10 Primex13 and 2 Primex41, Optitrack, Natural Point Inc.) have been selected. Both technologies have in fact reached maturity and are currently being implemented in the smart

industry ecosystem. Participants have been recorded while performing several static grasps, to cover the grasp taxonomy, and four different industry tasks: pick and place, object inspection, tool assisted assembly and hand assembly.

Such database, called HANDMI4, is now available on open access repository. From the industrial section of such database, a selected pool of statistical and costumed features were extracted and labelled to create an AI compatible dataset. Few machine learning-based classifier were trained and tested on HANDMI4, with a task classification accuracy reaching 90%.

The results of this research will contribute to the advancement of the field of HRC, and the open-access nature of the database will ensure that it can be utilized and built upon by future researchers and practitioners. This thesis provides valuable insights into the performance and limitations of the proposed solution and highlight the potential for future research and development in the field of motion recognition for Industry 4.0.

The remainder of the thesis is organized as follows. **Section 2** provides the state of art of MoCap technologies. **Section 3** illustrates the materials and the set up for the database creation. **Section 4** concentrates on the methodology used for the data collection, the database organization and the AI classification, while the results are shown in **Section 5** and then discuss in **Section 6**. An overview of possible future work is presented in **Section 7**.

Publications

- (1) Francesca Mongelli (2023). HANDMI4 (HAND Motion capture data for Industry 4.0) [Dataset]. <http://doi.org/10.21227/c6t8-ge47>
- (2) Mongelli, F., Menolotto, M., O'Flynn, B., & Demarchi, D. (n.d.). "Open Access Database of Industry 4.0 Tasks for the Development of AI-based Classifier." in SmartSystemIntegration (SSI), Bruges, Belgium. 2023 (Submitted)

2 State of Art

2.1 Hand

The hand is one of the most important tools we use to explore and interact with the surroundings. It has evolved to enable fine dexterity for object manipulation and sensing to perceived life-crucial physical phenomena, which we have used to shape our environments for thousands of years. We have a plethora of tools designed specifically to exploit the characteristics of our hand to interact with the surroundings [14]. The development of fine manipulation and hand dexterity was a crucial factor for the increase of the brain size and cortical surface during our evolution, promoting cognitive capability for tool usage [15]. ||

2.1.1 Hand's Anatomy

The human hand has 27 bones, which are controlled by 17 intrinsic muscles located within the hand and 18 extrinsic muscles located in the forearm. The hand contains several muscles and tendons, the former are the structures that may contract allowing the bones of the hand to move while the latter connect the muscles of the hand to the bone, allowing movement. Finally, there are the ligaments, fibrous structures that serve to hold the hand's joints together. This complex system can execute very fine and complex tasks for this reason, it has been a great source of inspiration for engineers and scientists in the development of tools, human-like robotic and prosthetic hands, who have developed several models to describe hand cinematic and hand motion skills [16].

The hand has three major types of bones including phalanges (14 bones in total), metacarpal bones (5 bones) and carpal bones (8 bones). Each finger is composed by three phalangeal segments (bones): distal-, middle- and proximal-phalange; except for the thumb which is composed by two phalangeal segments (as shown in **Figure 1**). The thumb is generally positioned at a different angle respect with the other fingers; in humans and most of the apes, it is rotated at the carpometacarpal joint, making it opposable to the other fingers and useful for grub firmly cylindrical

Commented [MM1]: I suggest to put a reference to the *Cortical Homunculus* here. Find one from an anatomy book. The one I gave you would be fine.

Commented [MM2]: I'm not sure I understand this. What is it?

Commented [MF3R2]: I meant to say that the evolution of human skill lies in product between hand and brain, so understanding of the tool (brain) and skill in using it (hand). But I know that written like this is very unclear.

Commented [MM4]: I would first specify how many natural joints there are in the hand, maybe also saying something about the fact that is difficult to give a precise number of DoF, since virtually all the bones' junctions act as joints. For the carpal bones you don't need to specify the number of joints, but you can say how many they are and that virtually every connections can be seeing as a small DoF.

Any information about the actual DoF are pure speculations based on conventions or approximations. In fact, many papers talk about 22DoF, other 23DoF, and again 24 or 25DoF.

and spherical objects and for picking up small objects in tandem with the other fingers [17].

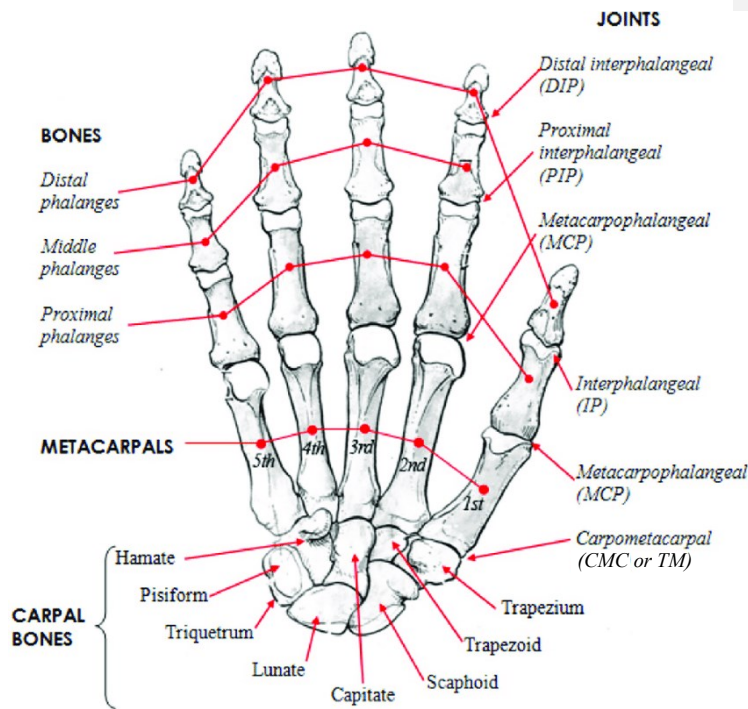


Figure 1. Representation of human hand with its joint and bones [10]

2.1.2 Biomechanical models for the hand

Kinematics is the field of study that studies the movement of bodies. It is used in mechanics and biomechanics to examine orientation, velocities, and accelerations (linear and angular) of rigid bodies without taking into account the forces that caused motion. In this part, we examine different kinematic models, which will be used to determine the orientation (also called pose) of each rigid segment on the hand (e.g., bones). In theory, a full and comprehensive kinematics model of the hand should detail all potential hand configurations. In practice however, the kinematic description of soft systems requires a very high level of complexity, which poses severe limitations from a computational prospective. For this reason, some simplifications are adopted in the field. The number of joints, their corresponding

degrees of freedom (DoFs) and ranges of motion (RoM) are limited, and generally differs from one model to the other for only few elements. The joints that we normally find for hand models (shown in **Figure 1**) in biomechanics are:

- Distal interphalangeal joint (DIP): it controls the relative pose between distal phalanges (DP) and PP. It has 1 DoF;
- Proximal interphalangeal joint (PIP): it controls the relative pose between MP and PP. It has 1 DoF;
- Metacarpophalangeal joint (MCP): the joint that connects the proximal phalange (PP) with the metacarpal one (MC). It has 2 DoF (abduction-adduction and flexion-extension)
- Interphalangeal joint (IP): the joint that constrains the pose of DP to the PP in the thumb. It has 1 DoF;
- Metacarpal phalangeal joint (MCP): the joint that control the flexion-extension of the thumb. It has 1 DoF;
- Trapeziometacarpal or Carpometacarpal joint (TM or CMC): the joint that control the flexion-extension, abduction-adduction, and the axial rotation of the thumb. It has 3 DoF;

It is simple to appreciate how complicated the human hand is, yet it is much more difficult to model. The hand may perform a wide range of movements and stances, but in order to study it thoroughly, you must first narrow down the number of DoFs. This 'number' varies depending on the sort of application for which the model is required. For example, in real-time applications, it would be ideal to reduce the number of DoFs to minimize the computing cost of the process, but if extreme precision is required, you must increase them by striking a balance. Following are examples of several models with varying degrees of freedom (DoFs).

2.1.2.1 Cobos' Model (6 and 9 DoF)

Cobos et al. [8], introduced a novel approach to cinematically describe the hand by hugely simplifying the model shown in **Figure 2**. They start from the traditional

Commented [MM5]: The reference goes here or at the end of the first sentence. Do the same also for the others.

Commented [MM6]: Debatable

24 DoF model and compare the possible hand configuration of such model (plotted using a [Cyberglove](#) [18]) with two new models, one with 6 DoF and the other with 9 DoF. Showing how the new models diverged from the original by a modest margin (10% for the former and 5%-10% for the latter).

Commented [MM7]: Reference here for Cyberglove

They achieved this reduction in DoFs by keeping only those fingers strictly necessary for task determination, thus considering only thumb, index, and little fingers, emphasizing how the remaining two fingers are less capable of independent movements, but rather they rely on the others. This offers several advantages from a computational point of view, as fewer DoFs to process and analyse can be crucial for real-time acquisitions where the model error is acceptable compared to the final goal.



Figure 2. Circular (on the left) and prismatic (on the right) grasping reconstruction. a) Original Gesture b) Hand model reconstructed by 9 elements c) Hand Model reconstructed by 6 elements [8]

2.1.2.2 Morecki's Model (22 DoF)

Morecki et al. [19] created a solution similar to the previous one employing a model with 22 DoFs for palm and fingers, and additional 3 DoFs located in the wrist and forearm. Their simplification is based on removing all joints with a range of motion minor than 5 degrees. In contrast to Cobos' model, the movements of the index and middle CMC are ignored because they do not significantly contribute to the mobility of those fingers. Instead, the CMC of the ring and the little are free to rotate (numbers 15 and 19 in **Figure 3**).

Commented [MM8]: Here the third and fourth are the middle and the index, right? To avoid confusion I will write the actual names of the fingers rather than the number, as sometime it's a bit ambiguous if the chosen model start to count the fingers from the pinkie or the thumb.

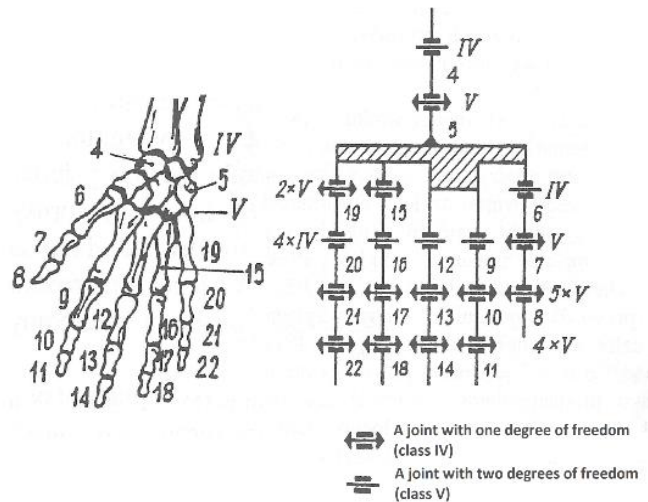


Figure 3. Representation of Morecki's model [11]

2.1.2.3 15 DoF's models

Several research works have been focusing on reducing the number of DoF of the hand model while keeping hand movements as accurate as possible. Among them, [20] and [5] have reduced the amount of DoFs by considering the hand's primary components and synergies. Observing actions like grabbing, for example, reveals that only a limited number of coordination patterns (synergies) that govern both joint mobility and strain of several fingers are required. Such constraints may be connected to biomechanical factors (synchronizations) between various motor neurons. Furthermore, statistical modelling of coordination patterns reveals that

a small number of so-called eigenpostures, or principal components (PCs), is adequate to reconstruct a vast number of hand motions. Moreover, a PC range was identified: lower-order PCs account for coarse movements such as closing and opening the hand, whereas higher-order PCs account for finer movements. The model chosen by the mentioned studies is depicted in **Figure 4** after various simplifications and assertions.

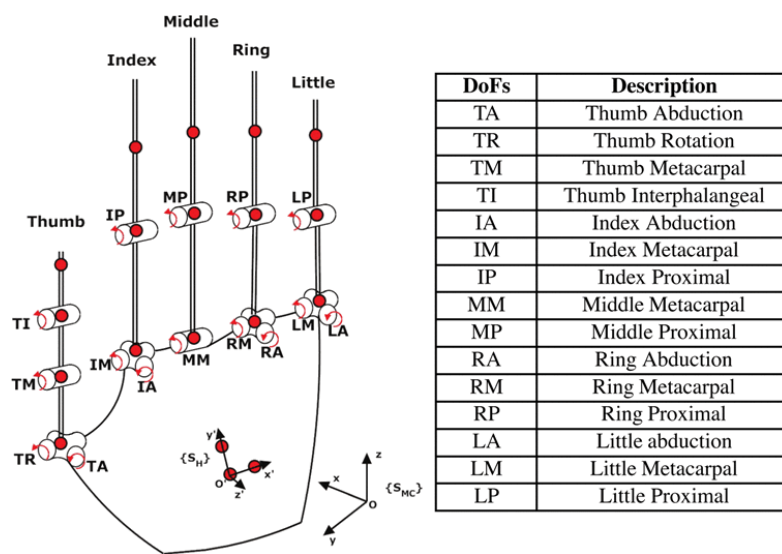


Figure 4. 15 DoF model's with the description of the DoF used [5].

2.1.2.4 23 DoF's model

As evidenced so far from the literature, the number of DoFs chosen for the hand biomechanical model is directly proportional to the level of complexity of the tasks such model is meant to describe. The final hand model presented has been used to develop wearable devices with hand gesture recognition for telemedicine applications [21] and to create a glove with tactical feedback for hand tracking [11]. **Figure 5** depicts a 21 DoF model with 4 DoFs for each finger (2 DoFs in MCP and 1 DoF in PIP and DIP) except for the thumb, which has 5 DoFs (2 DoFs in TM, 2 DoFs

Commented [MM9]: It seems that the figure 5 has only 21 DoF. Could you double check please.

in MCP and 1 in IP). It's easy to observe how fine hand modelling is required in such applications as the level of dexterity involved is high.

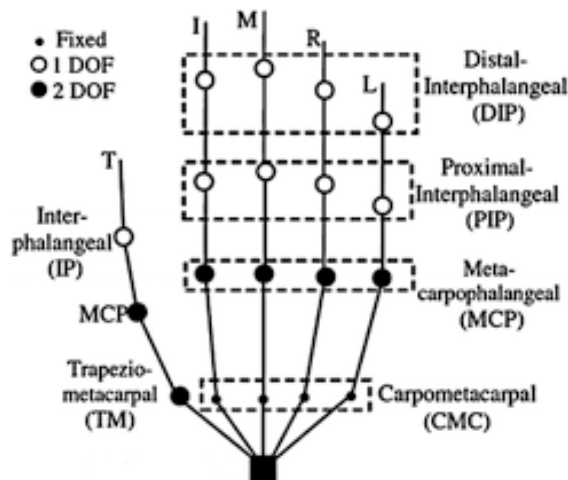


Figure 5. A 21 DoF's model [7]

2.2 Motion Capture

When attempting to define human motion, the first element of valuable information is the kinematics, or how the body moves in space in terms of position, velocity, and acceleration. Over the years, research in the field of motion analysis has largely improved, exploiting the progressive downsizing of the sensing technology, in combination with more powerful and sophisticated tracking algorithms, gradually reducing measurement errors [22].

From hereafter, we will refer to human body MoCap as the ability of a sensing technology to detect the position/orientation of one or many rigid bodies, usually body segments (e.g. a finger can be schematized as three rigid bodies (distal, middle and proximal phalangeal) connected by two joints with 1 DoF each), to some global coordinates, or the relative position/orientation of one rigid body to another one (or to the global coordinates).

MoCap technologies can be divided into two categories: optical systems, which primarily use cameras that can operate at different spectrum ranges, and non-optical systems, which exploits sensors that use other physical principals to track

Commented [MM10]: Post-processing here is a bit vague.

Commented [MM11]: I added primarily because camera system are NOT the only optical MoCap system. Optical fibers and Bragg grating are other optoelectronic technologies.

motion. The first, in general capture human body motion by using two or more cameras to triangulate the location and/or orientation (or pose) of specific body landmarks, known as '*anatomical land markers*', relative to the cameras' coordinate system. The non-optical method uses, for instance, the intensity (or the orientation) of magnetic fields generated by inductive coils placed on the rigid body under observation, or again the displacement of microscopic spring loaded masses (accelerometers or gyroscopes), and so on [23].

A table explaining the pros and cons of each system (**Table 1**) is provided at the end of the chapter.

2.2.1 Camera based system.

MoCap systems that uses cameras can either directly track the object under examination or track specific markers placed on the object. The last option usually improves contrast and tracking performance in general. Whether such markers are actively producing light or passively reflecting light, the camera systems are equipped with image sensors designed to detect a specific range of the light spectrum. In the case of passive markers, the camera itself also mount the emitting source of light (generally infrared (IR) LED).

The cameras work thanks to the principle of triangulation, shown in **Figure 7**, that is a method of determining the 3D position of a point in space by using the images of that point taken from two or more cameras. The principle behind camera triangulation is based on the fact that the object or point is viewed from different angles, the images captured by each camera will be slightly different due to the change in perspective. To determine the 3D position of a point, the two or more camera positions and their respective images of the point are used to create two or more lines of sight, also known as rays. The point in 3D space that these rays intersect is the 3D position of the point being observed. This intersection point can be calculated using principles of trigonometry and linear algebra.

Commented [MM12]: In table 1 there is no section dedicated to the use of ultrasound sensors.

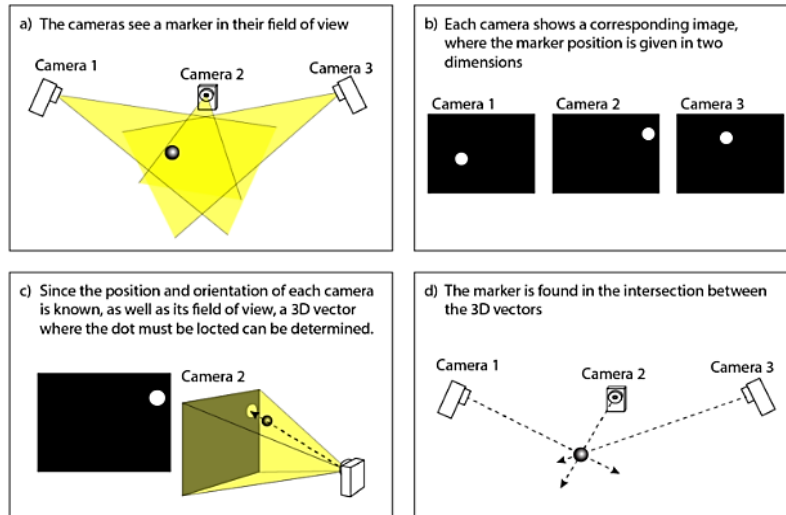


Figure 7. The principle of camera triangulation.[4]

In order to perform camera triangulation, the cameras must be calibrated, which is the first step in the process of data acquisition (Figure 6).

Commented [MM13]: I would also appreciate a picture where you can clearly see the principle behind the camera triangulation. Something like this:

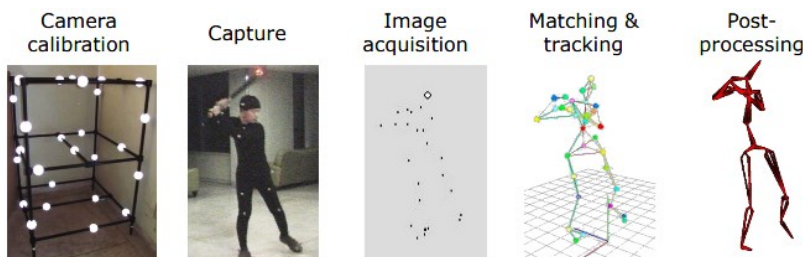


Figure 6. Optical motion capture process

The first step is the camera system calibration. With this, the relative distance between each camera is determined, which is then used by the triangulation algorithm to establish the position of each marker with respect to the camera coordinate system. To achieve such calibration, a reference object, which mounts markers placed at a known distance from each other (usually called *wand*), is waved in front of the cameras repeatedly throughout the capturing volume. Here the ground plane is also established, together with origin of the camera coordinate system. Secondly, the body segment of the subject under examination is prepared by placing the markers (spherical or semi-spherical objects with different sizes, from

some centimetres to few millimetres) on specific points of interest. Such points can be body landmarks or just distributed around the body segment so that in any orientation at least three markers are visible by at least two cameras. In fact, three markers are sufficient to effectively determine the plane containing the body segment. In the next step, using the MoCap software, each marker is either associated with a specific body landmark either as single marker or clustered together as part of the same rigid body. Finally, the motion data acquisition is carried out. After the data acquisition, there are typically several post-processing steps, which usually help the motion tracking algorithm to resolve the markers trajectories, filtering noise, avoiding outliers or marker mislabelling.

The optoelectronic system is able to reconstruct the orientation of the body segment and the 3D position of each individual marker, or the rigid body (markers cluster). The spatial resolution of a camera system is determined by the size of the measuring field. A wider field reduces the spatial resolution of each individual camera. The maximum resolution of a CMOS image sensor depends on the distance of the object being observed, the field of view (FOV), and the number of pixels in the axis being considered, reported in the equation (1).

$$MaxRes_Y = \frac{D \cdot \tan^{-1}(FOV_Y/2)}{N_{pixel_Y}} \quad (1)$$

For example, an image sensor with a resolution of 1024x1280 pixels and an optical system allowing for a FOV of 56 degrees at a distance of 1 meter can resolve a maximum of 2.93 mm per pixel in the Y-axis.

This formula represents the maximum resolution for a single camera. Using multiple cameras can improve the overall resolution of the system, with each camera contributing to the final image (Equation(2)). However, the resolution of each individual camera will still be limited by the field of view and distance to the object being observed. As a rough estimation, we can assume the resolution of a system with N cameras be:

$$Res_Y \cong \frac{MaxRes_Y}{N} \quad (2)$$

The main limitation of these systems is occlusion, i.e., an object placed between the marker and the camera, which prevents the camera from having a clean line of sight of the object under examination, effectively preventing its tracking. This problem can be mitigated by increasing the number of cameras and their positioning around the tracking volume.

2.2.1.1 Passive Markers

Passive markers (examples shown in **Figure 8**) are spherical or semi-spherical objects covered with high reflecting material (usually in the IR spectrum). They can be applied directly onto the points of interest using adhesive materials (i.e., double-sided tape, glue, etc.). They are more frequently employed in controlled environment, such as laboratories and recording studios.



Figure 8. Different size of passive markers [3]

2.2.1.2 Active Markers

Active markers are composed by light emitting source, mostly a LED, and a power supply enclosed in a compact unit (**Figure 9**). They are less frequently used than passive markers, but they represent the best solution for outdoor measurements, where the environmental light and reflections are not as controlled as in the laboratory. Two are the advantages of active markers: they are more visible in environment with uncontrolled lighting and lot of natural reflections and makes markers mislabelling less likely.

Commented [MM14]: They are co-protagonist really. So no need for this sentence. Also a bit too informal and subjective.

Commented [MM15]: Move this after the Passive Markers section

In fact, active markers can be set to different wavelengths or different blinking frequencies, or be multiplexed one at a time, highlighting only certain areas or activating only in front of certain cameras. The disadvantage of multiplexed markers is that the sampling rate is divided by the number of markers used thus sacrificing the frame rate.

Commented [MM16]: What's the disadvantage here? Probably that you need to sacrifice in frame rate. Correct?



Figure 9. An example of active markers [2]

2.2.1.3 Marker-less

A marker-less MoCap example is shown in **Figure 10**. It is rarely employed in clinical settings due to large measurement errors; nonetheless, it is widely used in the film industry. The main reasons are that it works using the visible spectrum of the light and without any markers on the subject. Here as well multiple cameras are used to better estimate the figure's 3D silhouette. The entire effort is focused on

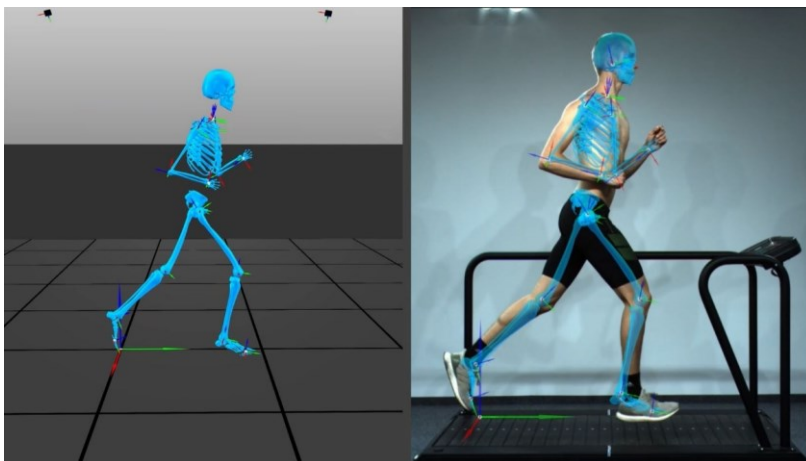


Figure 10. An example of marker less MoCap [1]

processing the acquired frames and is based on image segmentation and identification of human poses that can match some pre-existing model [24, 25].

2.2.2 Non camera-based systems

2.2.2.1 Electro-mechanical

Electromechanical systems for tracking are based on the use of sensors, which can detect changes in position and orientation, and then translate those changes into electrical signals that can be processed and used to control mechanical devices. The most commonly used sensors in electromechanical tracking systems are optical sensors and potentiometer.

Optical sensors use light to detect changes in position and orientation. These sensors work by projecting a beam of light onto a target, and then measuring the reflected light using photodetectors. The position and orientation of the target can be determined based on the position of the reflected light relative to the original beam. The principle behind a potentiometric sensor is based on the relationship between resistance and position. A potentiometer consists of a resistive element and a wiper arm that moves along the element. As the wiper arm moves, the resistance between the wiper and the end terminals changes. The position of the wiper arm can be measured based on the resistance between the wiper and the end terminals [6].

Their field of application is restricted as they are bulky and heavy structures that restrict movements. A visual example is given in **Figure 11**.

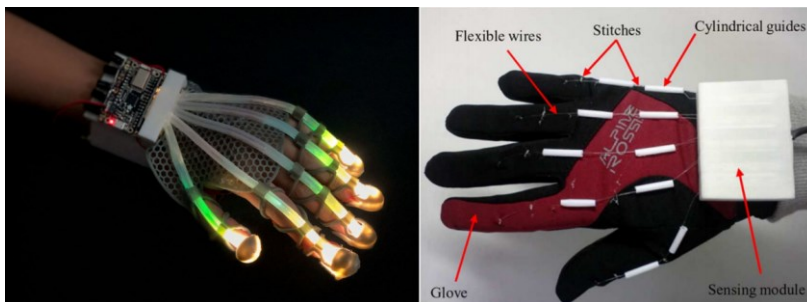


Figure 11. Electromechanical systems. On the left an example of optical fibre while on the right a potentiometer one [6].

2.2.2.2 Magnetic

Electromagnetic tracking is a method for measuring the position and orientation of an object or system using electromagnetic fields. The principle behind electromagnetic tracking is based on the use of electromagnetic sensors, which can detect changes in position and orientation and translate those changes into electrical signals that can be processed to determine the position and orientation of the object or system being tracked.

In an electromagnetic tracking system, a transmitter generates a magnetic field, while one or more sensors detect the field and calculate the position and orientation of the object or system based on the strength and direction of the magnetic field. The sensors typically consist of a coil or series of coils that generate electrical signals in response to changes in the magnetic field. The value that electromagnetic tracking measures is the position and orientation of the object or system being tracked. This information is typically expressed in terms of the x, y, and z coordinates of the object or system, as well as its pitch, yaw, and roll angles.

The intensity of the magnetic field in an electromagnetic tracking system can vary depending on the application and the design of the system. Typically, the field strength ranges from a few mT to several Tesla. The field strength required for a given application depends on factors such as the distance between the transmitter and sensor, the size and shape of the object or system being tracked, and the required accuracy and precision of the tracking system.

The performance of electromagnetic systems depends on the cost of the system and the resolution in general depends on the distance of the sensors from the source, which is why it is a popular system for hand tracking [26, 27] cause you can keep the hand near the source and perform fine tracking [9]. For the same reason, it is not comfortable for a gait analysis. However, there are some limitations to electromagnetic tracking. For example, the accuracy and precision of the system can be affected by the presence of other magnetic fields in the environment, such as those generated by electronic equipment or the Earth's magnetic field.

Commented [MM17]: I didn't get why is popular. Because it's very cheap? It's also very much affected by ferromagnetic interferences and other sources of noise, so it's a good compromise only in very specific applications.

A visual example is reported in **Figure 12**.

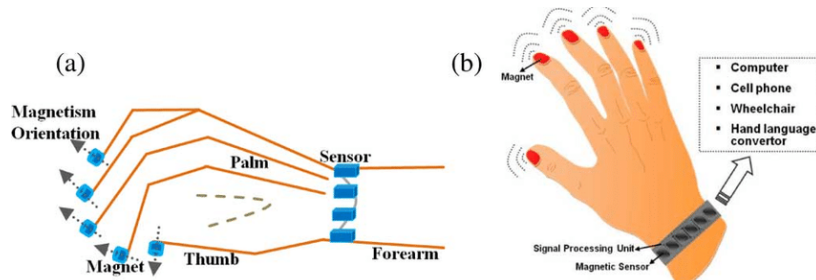


Figure 12. An example of magnetic tracking system [9]

2.2.2.3 Inertial

Inertial tracking is a method for measuring the position and orientation of an object or system using inertial sensors, such as accelerometers and gyroscopes. (**Figure 13**) The principle behind inertial tracking is based on the fact that when an object moves, it experiences changes in acceleration and rotation, which can be measured by inertial sensors. By integrating these measurements over time, the position and orientation of the object can be determined.



Figure 13. The Tyndall data glove, with IMU inside, use in the data acquisition of this study.

In an inertial tracking system, accelerometers measure changes in acceleration along the x, y, and z axes, while gyroscopes measure changes in rotation around

these axes. The values that inertial tracking measures include linear acceleration and rotational velocity, as well as the object's position and orientation over time.

They can be implemented using different types of sensors, such as micro-electro-mechanical systems (MEMS) accelerometers and gyroscopes. In particular, a linear accelerometer measures linear acceleration by detecting the displacement of a proof mass relative to a fixed frame, while an angular accelerometer or gyroscope measures angular velocity (i.e., rate of change of angle) by detecting the Coriolis force that acts on a vibrating proof mass (**Figure 14**). MEMS sensors have revolutionized everyday life by enabling the creation of small and low-power sensors that can be integrated into wearable devices and other mobile platforms. Thanks to their compact size and low power consumption, MEMS sensors have made it possible to incorporate sophisticated sensing capabilities into a wide range of products, including fitness trackers, smartphones, and IoT devices. This has led to new opportunities for monitoring and understanding the world around us and has opened up new possibilities for the development of innovative and disruptive technologies.

One advantage of inertial tracking is that it does not require external references, such as cameras or electromagnetic fields, to operate. This makes it suitable for use in environments where other tracking methods may not be practical or effective, such as in outdoor environments or in areas with limited space.

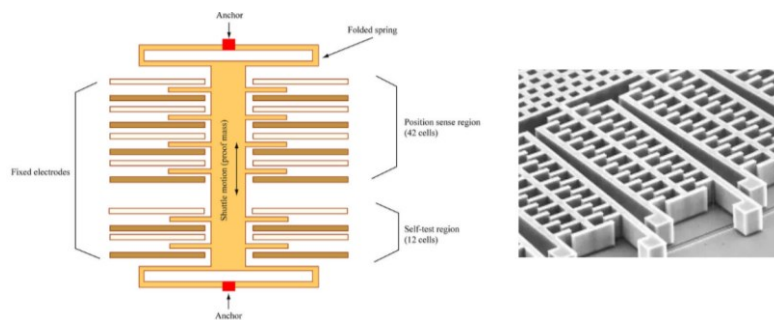


Figure 14. An example of MEMS sensor. [12]

However, there are some limitations to inertial tracking. For example, the accuracy and precision of the system can be affected by sensor drift, noise, and bias, which can accumulate over time and result in errors in position and orientation calculations. In addition, inertial tracking is generally most accurate over short periods of time and can suffer from long-term drift, which may require periodic recalibration to maintain accuracy.

So, within each inertial sensor the information provided by the different sensors is integrated using a filter (more info in the lists below), which weights the estimates provided by the sensors differently providing information on the orientation of the inertial sensor as a whole.

For the sake of clarity here's a list of some of the most common pose estimation algorithms:

1. **Complementary Filter:** This is a simple algorithm that fuses accelerometer and gyroscope data to estimate the roll and pitch angles of an object. It uses a low-pass filter to integrate the accelerometer data and a high-pass filter to integrate the gyroscope data and combines the two to produce a more accurate estimate of the object's orientation.
2. **Kalman Filter:** The Kalman filter is a more advanced algorithm that uses a statistical model of the system dynamics and measurement noise to estimate the state of the system. It can be used to estimate the roll, pitch, and yaw angles of an object based on data from accelerometers and gyroscopes and is particularly useful for reducing errors caused by sensor drift.
3. **Extended Kalman Filter (EKF):** The EKF is an extension of the Kalman filter that can handle nonlinear system models. It is commonly used in applications where the dynamics of the system are nonlinear, such as in robotics and navigation systems.
4. **Unscented Kalman Filter (UKF):** The UKF is a further extension of the Kalman filter that uses a nonlinear transformation of the state variables to better model the system dynamics. It is particularly useful in applications

Commented [MM18]: You didn't mention the fact that MEMS have revolution the world of portable devices and wearables, as they are tiny, low energy consumption and cheap, and can easily be integrated in small devices or garments.

where the system dynamics are highly nonlinear, such as in robotics and aerospace.

5. **The Gauss-Newton filter:** is a nonlinear optimization algorithm used for estimating the pose of an object based on sensor data from an IMU. It minimizes the difference between the predicted sensor measurements and the actual sensor measurements, using a least-squares approach. It's particularly useful for highly nonlinear systems but can be computationally expensive. It's often used in combination with other algorithms, such as the Kalman filter or complementary filter.

2.2.3 Features of each technique

Here is a table showing the advantages and disadvantages of the different systems.

Table 1. Pros and Cons of the systems analysed above.

	Method	Pros	Cons
CAMERA – BASED	Optical – Passive	<ul style="list-style-type: none"> • Precision (<1mm) • Wireless • No power supplied needed. 	<ul style="list-style-type: none"> • Occlusion • Position only • Manual clean-up after captures. • Post processing latency • Limited measurement space
	Optical – Active	<ul style="list-style-type: none"> • Precision (<1mm) • Wireless • Better than passive at finding correspondences. • Higher range than passive 	<ul style="list-style-type: none"> • Position only • Occlusion • Post-processing latency • Sample rate divided among sensor
	Optical – Marker less	<ul style="list-style-type: none"> • Flexible • No sensor needed. • Wireless – outdoor 	<ul style="list-style-type: none"> • High noise • Occlusion • High sensitivity to lighting
NON CAMERA – BASED	Electro-mechanical	<ul style="list-style-type: none"> • Truly real-time (500 Hz) • High accuracy • Inexpensive • No occlusion 	<ul style="list-style-type: none"> • Restriction of movement • Needs to match body proportion. • No global position

	Magnetic	<ul style="list-style-type: none"> • Measure position and orientation in 3D • Low encumbrance • Cost • Good performance close to emitter 	<ul style="list-style-type: none"> • Accuracy affected by metal objects. • Operate on only one side of the source. • Low range • Calibration
	Inertial	<ul style="list-style-type: none"> • Wide area • Inexpensive • Orientation very accurate • Minimal interference • Encumbrance 	<ul style="list-style-type: none"> • Position poor • Calibration • Inaccurate over time • Drift

2.3 Data Gloves

With the term data glove, we refer to sensing devices that can be worn like gloves, which provide some form of information regarding the hand pose by streaming data either via wires or wirelessly. Over the last 40 years, data gloves have been developed to facilitate human-computer interaction based on hand and finger movement. Despite the ongoing development of applications that take advantage of data gloves, such as virtual reality [28], teleoperation [29], remote control of robotics [30], and video games [31], these have yet to enter the mainstream. One reason for this could be that does not obstruct finger movements and adaptation to satisfy completely, but also non-functional specifications such as wearability that does not obstruct finger movements and adaptation to different hand sizes [13].

It is noteworthy to notice how the terminology regarding data glove is wide in the literature, with manes that can refer to gloves able to track pose but also to transmit haptic feedback, such as: "cyber gloves", "haptic gloves", "virtual gloves", "data gloves", "smart gloves" or "sensory gloves". In general, "date glove" defines a glove capable of acquiring finger and hand movement or pose for motion tracking purposes, using some sensing technology embedded in the glove. Similarly, "haptic gloves" are generally referred to devices able to provide tactile feedback via some actuators placed on specific locations. In this work, we will use the general term "smart glove" to refer to such devices in a broad sense.

Commented [MM19]: Is this not still within the wearability problem? Maybe you want to add the ergonomics or usability issue here.

It might be worth it here to explain the concept of wearability (and maybe also ergonomics, usability and washability, which is important in wearable devices). It will be useful for the discussion where we will highlight the pros and cons of the glove.

2.3.1 Glove types

Smart gloves on the market are typically divided into two categories: exoskeleton and fabric, though looking at the **Figure 15**, it is clear how other significant elements can identify different types of gloves, such as open structure, the material and so on.



Figure 15. Pictures some examples of smart gloves. [13]

2.3.2 Capability

Smart gloves are versatile devices that have applications in a wide range of fields, including virtual and augmented reality, rehabilitation and physical therapy, sign language translation, robotics, and gesture recognition. These fields rely on the two main functions of hand and finger tracking and haptic feedback.

The latter is used to simulate touch through the use of the glove and can provide two sorts of feedback: kinesthetics and haptic. Kinesthetics feedback simulates movement by replicating force and resistance via electrical or mechanical actuators that provoke sensations such as weight or inertia. When attempting to accomplish this, it is critical to have good force resolution and so that even minor changes may be detected and a low latency for the real time. The goal of haptic feedback is to recreate the perception of touch, such as surface roughness, shape, pressure, or temperature.

Data gloves, on the other hand, use sensors embedded in the glove to measure the orientation of the fingers and hand, from which important information can be extracted, such as the joint angles between two bone segments, the relative acceleration between two fingers, or qualitative information to analyse a subject's range of motion. The Tyndall Smart Glove, which is explicitly described in section 3.2, is utilised in this study. It is an open structure data glove which uses IMUs to track fingers, the palm, and the wrist, allowing the evaluation of the hand and fingers pose.

2.3.3 Wearability

The wearability of smart gloves is one of the most crucial factors to consider. Wearability refers to the ease and comfort of wearing the glove for an extended period of time, as well as its ability to perform its intended functions effectively. This concept is particularly important because the glove must be comfortable enough to wear for an extended period of time, while also allowing the user to perform their intended tasks. This means that the glove should fit well, be made from breathable materials, and not impede the movement of the wearer's hands.

Commented [MM20]: That's pretty much the only function of such devices :D

Commented [MM21]: And low latency, in fact the real crucial aspect in haptic feedback is to trick the brain that what you are seeing and what you are touching are the same thing.

Commented [MM22]: Sorry I see only now you have a section on this.

Washability is also an important factor to consider when designing a smart glove, particularly if the glove is intended for long-term use or use in a clinical or industrial setting. Ideally, the smart glove should be designed to allow for easy cleaning and maintenance, without damaging the electronic components or sensors. In addition, the sensors and other electronic components within the smart glove must be designed in a way that does not interfere with the glove's wearability. For example, they should not add unnecessary bulk or weight to the glove, and they should not be positioned in a way that creates discomfort or limits movement.

The device's size, weight, and energy consumption are critical features to examine. The size should be adaptable to any hand, which means that the device is either modular (composed of parts that can adapt to different lengths) or made in multiple sizes. The weight of these gadgets should be between 50 and 300 g on average to allow for seamless operation without fatiguing the operator. Finally, in terms of battery life, it is important to guarantee adequate battery autonomy according to the duration of the task under examination; the autonomy of these devices ranges between 2 and 10 hours on average, depending on battery capacity and energy consumption of the device [13].

2.4 Open Access Database

The availability of open access databases for Human Activity Recognition (HAR) has seen a substantial growth in recent years due to the increasing prevalence of wearable sensors and other MoCap technology. Wearable sensors, such as smartwatches, fitness trackers, and clothing with embedded sensors, generate a large amount of data that can be used to classify a wide range of tasks, including physical activities, sleep patterns, and falls. Such open access databases are becoming increasingly attractive for researchers and practitioners in several fields, such as human computer interaction, biomechanics, and rehabilitation [32, 33].

One of the key developments in the field is the use of machine learning algorithms for task classification. These algorithms can be trained on large amounts of data generated by wearable sensors, allowing for accurate classification of tasks. Some of the most commonly used algorithms include support vector machines (SVMs),

Commented [MM23]: It would be nice you can find the proper definition of wearability for wearable devices (if it exist).

decision trees, and deep neural networks. The classification performance of these algorithms are linked to the availability of large databases, which provide the necessary data for training and testing [32, 34].

In addition to machine learning algorithms, cloud-based platforms have become increasingly important in the field of open access databases for task classification. These platforms allow for easy sharing and collaboration among researchers, as well as providing access to large amounts of data. The use of cloud-based platforms also enables researchers to store and analyse data from a variety of wearable sensors, making it possible to compare results across different devices and applications [34].

Another important aspect of the state of the art of open access databases for task classification is the growing focus on the ethical and privacy implications of collecting and using data from wearable sensors. There are concerns about the use of personal data, particularly in the fields of healthcare and rehabilitation, where sensitive information is often collected. To address these concerns, many open access databases have implemented strict privacy policies and secure storage methods to protect the privacy [35].

Overall, the creation of open access databases is increasing, with new MoCap technologies and approaches being developed and tested. Such trend towards making datasets open access is likely to continue, particularly as wearable sensors become more pervasive and the demand for data grows. As the field continues to evolve, it will be important to balance the benefits of open access databases with the need to protect personal data and ensure ethical use of this information.

Here an overview of the existing open access database for the hand motion tracking [36]:

Table 2. Overview of available open access dataset in the context of hand gesture.

Year	Dataset name	Ref. Dataset	Ref. Paper	Format	No Subjects	No Activities	Sensors (sample rate)
2009	HCI gestures	[37]	[38]	mat	1	5	IMU (96 Hz)
2010	KIT Whole-Body Human Motion Database	[39]	[40]	xml, c3d, avi	224	43	RGB
2012	OPPORTUNITY Activity Recognition Dataset	[41]	[42]	txt	12	24	IMU (100 Hz)
2013	Hand Gesture	[43]	[44]	mat	2	11	IMU (32 Hz)
2015	Skoda Mini Checkpoint	[37]	[45]	mat	1	10	IMU (98 Hz)
2015	UTD Multi-modal Human Action Dataset (UTD-MHAD)	[46]	[47]	mat, avi	8	27	IMU (50 Hz), depth camera

2.5 AI classifier

In recent years, HRC has received considerable attention as an emerging technology in academia and industry. In HRC assembly, for example, robots are frequently required to dynamically change their pre-planned trajectories and control parameters in order to interact with humans in a shared workspace. In contrast, today's industrial robots are still controlled by pre-generated rigid algorithms that cannot support successful HRC that requires real-time adaptability. Human motion prediction, for example, is critical for both collision avoidance and proactive aid of robots to people, in addition to multi-modal robot control and in-situ operator support, in response to the requirement for greater adaptability. Deep Learning (DL) and Machine Learning (ML) have been shown to be effective for classification, recognition, and detection of context awareness.

There are three subfamilies of learning approaches we can identify in the literature: supervised, unsupervised, and reinforcement. A graphical representation is offered in **Figure 16**. [48].

Supervised Learning (SL) is a type of ML in which a model is trained on a labelled dataset, where the correct output is already provided. The algorithm then learns to make predictions by finding relationships between the input and output variables. Examples of supervised learning include linear regression, k-nearest neighbours, decision trees, and neural networks.

Commented [MM24]: A bit vague

Unsupervised Learning (UL) is a type of ML in which the algorithm is not provided with labelled data. Instead, the algorithm must find patterns and structure in the data on its own. The goal of unsupervised learning is to discover hidden structures or patterns in the data without any prior knowledge of the output. Examples of unsupervised learning include k-means clustering, principal component analysis, autoencoders, and hierarchical clustering.

Reinforcement learning (RL) is a type of ML in which an agent learns to make decisions by interacting with its environment and receiving rewards or punishments. The agent learns to take actions that maximize the cumulative reward over time. Reinforcement learning is different from supervised and unsupervised learning in that it focuses on decision-making rather than prediction. The agent is trained to take actions in order to maximize a reward signal. Examples of reinforcement learning include Q-learning, SARSA, and deep reinforcement learning.

In this thesis the goal is to classify tasks previously recorded with MoCap techniques, it was therefore decided to explore the various methods of supervised learning having already labelled data.

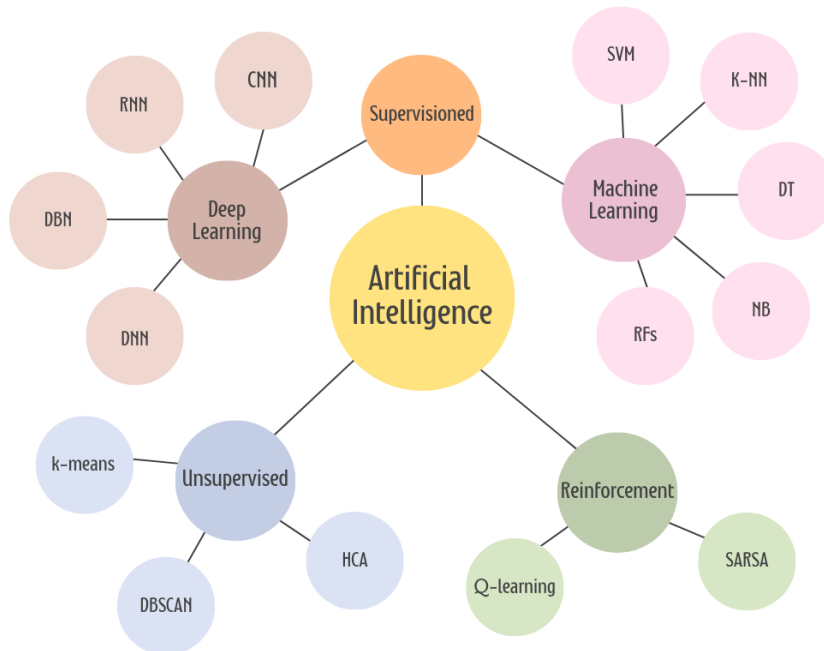


Figure 16. AI taxonomy

2.5.1 Typologies of SL

There are two main algorithmic strategies for SL: classification and regression. In classification, the output variable is categorical, and the algorithm tries to learn which category a new example belongs to, based on the input variables. In regression, the output variable is continuous, and the algorithm tries to learn the relationship between the input variables and the output variable, so that it can predict the value of the output for new examples. SL is widely used in many applications, such as image recognition, speech recognition, natural language processing, and predictive analytics. The performance of SL algorithms depends on the quality of the training data, the choice of features, and the choice of algorithm.

There are several SL methods that are commonly used for task recognition, including:

- **Convolutional Neural Networks (CNNs):** is a type of DL algorithm that is commonly used for image and video classification tasks. It is called "convolutional" because it uses convolutional layers, which apply a convolution operation to the input data. The basic building blocks of a CNN are convolutional layers, activation layers, pooling layers, and fully connected layers. In a classification task, the CNN's output is a probability distribution over the classes. The CNN is trained to minimize a loss function, such as cross-entropy, that measures the difference between the predicted probability distribution and the actual class labels. The backpropagation algorithm is used to update the model parameters and minimize the loss. Overall, CNNs are a powerful tool for task classification and have achieved state-of-the-art results on many benchmark datasets in computer vision and related domains [49-51].
- **Recurrent Neural Networks (RNNs):** is a type of deep learning algorithm that is commonly used for tasks involving sequential data, such as speech recognition, natural language processing, and time series prediction. Unlike feedforward neural networks, RNNs have a loop that allows information to be passed from one step of the sequence to the next. For task classification, an RNN can be used to classify sequences of data, such as speech signals or text. They take in a sequence of input data, processes it step by step, and outputs a probability distribution over the classes at each time step or at the final time step. They are also capable of handling variable-length sequences, which is important for many real-world applications [52, 53].
- **Support Vector Machines (SVMs):** are a type of machine learning algorithm that are commonly used for classification tasks. The goal of an SVM is to find the optimal boundary that separates the classes in the feature space. This boundary is called a hyperplane and is defined by a weight vector and a bias term. The hyperplane is chosen so as to maximize the margin, which is the distance between the hyperplane and the closest data points from each class.

These closest data points are called support vectors and are the key to determining the hyperplane. SVMs are well-suited to classification tasks where the data is linearly separable or can be transformed into a feature space where it is linearly separable. They are also effective when there are only a few training examples, as they are less prone to overfitting than other methods that rely on complex models.

- **Random Forests (RF):** These are a type of ensemble learning method that can be used for both classification and regression tasks. The basic idea behind random forests is to create a collection of decision trees and combine their predictions to obtain a final prediction. A decision tree is a tree-like structure where each node represents a test on one of the input features, and the branches represent the possible outcomes of the test. The leaves of the tree contain the class predictions. To create a random forest, multiple decision trees are trained on randomly selected subsets of the training data and features. The final prediction is obtained by taking a majority vote of the predictions made by each tree. Random forests are well-suited to classification tasks where the data has complex relationships between the input features and the classes, and where interpretability of the model is important. They are also effective when there are many input features, as they can automatically handle feature selection [54, 55].

3 Materials

3.1 Camera-based Equipment

The MoCap system used in this study consists of 12 Optitrack cameras (NaturalPoint, Inc. DBA Optitrack, Corvallis, Oregon, US) of which ten are PrimeX13 and two PrimeX41, their main features are described in **Table 3** and a visual example is depicted in **Figure 17**.

Table 3. Camera-based specs' system [56].

Features	Prime ^x 13	Prime ^x 41
Resolution	1280 x 1024	2048 x 2048
Native Frame Rate	240 Hz	180 Hz
Latency	4.2 ms	5.5 ms
3D Accuracy	± 0.20 mm	± 0.10 mm
Passive Marker Range	16 m	30 m
Stock Lens	56 x 46 FOV (5.5mm)	51 x 51 FOV (12 mm)
LED Ring	10 LEDs	20 LEDs
Size	(6.86 x 6.86 x 5.46) cm	(12.6 x 12.6x 13.2) cm
Weight	0.32 Kg	1.36 Kg
Cost	\$2.499	\$6.499

The data was collected at a frame rate of 240 Hz and exposure time (time interval in which the light is let to hit the image sensor) of 20μs. Such Exposure time was



Figure 17. PrimeX13 camera on the left and PrimeX41 on the right

Materials

chosen as a good compromise between ensuring highly reflective objects (in our case the reflective markers) were clearly captured and preventing small reflections and other noise sources to be part of the data acquisition.

To track each anatomical segment of the hand, an existing biomechanical model implemented on the MoCap software, Motive (by Optitrack [57]), was used. For this model 10 reflective markers were placed on the hand, specifically 5 markers of 9mm diameter placed on the palm section, placed on a black non-reflective glove, and 5 markers of 6mm diameter were placed on each nail. The use of smaller markers on the nails minimises the overlap of reflected signals during the data acquisition, hence minimising marker mislabelling.

To incorporate the forearm in the model, 4 markers of 14mm diameter were placed on an elasticated binding and then wrapped around the subject's forearm as shown in **Figure 18** below.



Figure 18. Placement of markers on the hand

3.2 Smart Glove Equipment

The Tyndall smart glove contains twelve 9-axis IMUs (each with a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer) strategically located to account for the degrees of freedom of each finger joint of the hand - two per finger, one on the palm's back, and one on the wrist, a visual example is given in **Figure 19** [58]. To monitor orientation and biomechanical characteristics, IMUs are placed on the flexible connection and on the phalange of each finger segment. The MPU-

Commented [BO25]: Link? Reference?

Commented [BO26]: Diameter?

Commented [BO27]: check

Commented [BO28]: Reference !

Materials

9250 sensor has a high-sensitivity tri-axis angular rate sensor (gyroscope) with a full-scale range of from ± 250 to ± 2000 dps, a tri-axis accelerometer with a programmable full scale range from $\pm 2g$ to $\pm 16g$, and a tri-axis compass with a full scale range of $1200\mu T$. In this research, we decided to use only three fingers out of five (thumb, index, and middle) to improve device wearability and minimise the discomfort caused by the interaction between the glove and the object manipulated during the trial.



Figure 19. Tyndall Smart glove

3.3 Lab setup

To minimize subject movement artefacts impacting on the model accuracy, the entire data acquisition was performed with the subject in a seated position. The centre of the motion capture volume incorporates a table, with minimal frame, and a chair. This setup simulates as closely as possible a classic industrial working setting [59]. The twelve cameras of the MoCap system were placed all around the motion capture volume, which for the purposes of this experiment was considered to be one cubic meter with base of this volume at the table's top level (Figure 21). The final arrangement of the cameras is the result of many empirical experiments carried out to optimise the experimental setup so as to minimize object obstruction and optimize image resolution. Initially, the cameras were placed on a rigid

Commented [MM29]: First the equipment and then the setup. Move this after the equipment section.

Commented [BO30]: Not sure what you mean here

Commented [MF31R30]: I meant that don't take much space

Commented [BO32]: reference

Materials

rig made of aluminium profiles, pointing towards the centre of the structure itself (the capturing volume), shown in **Figure 20**.

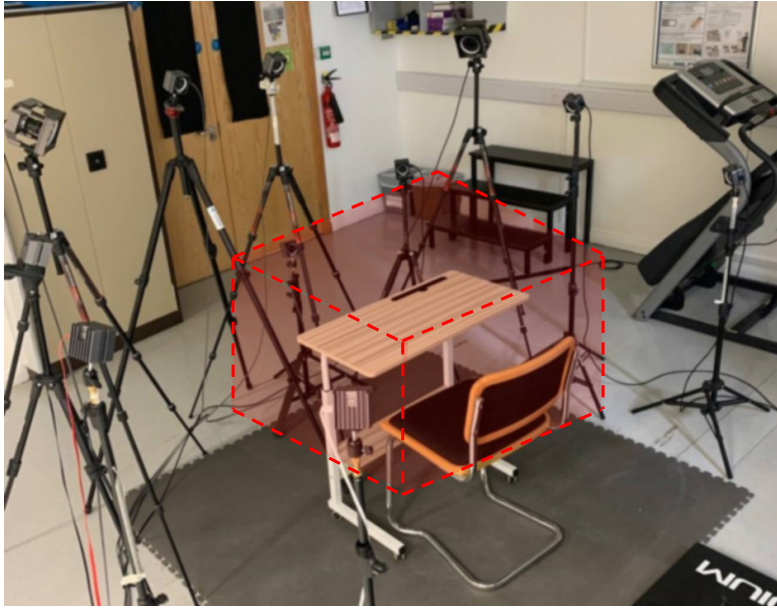


Figure 20. Final lab setup for the study, in the red cube the shooting volume (1mx1m)

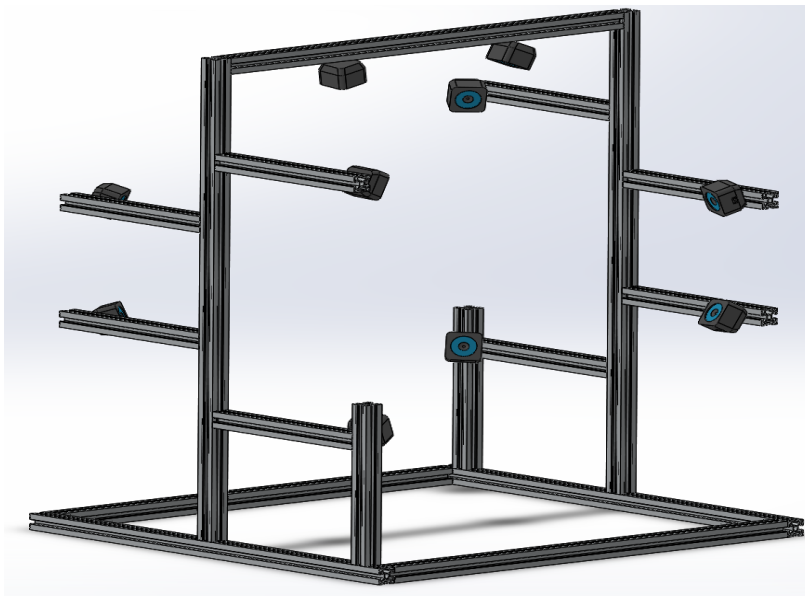


Figure 21. Aluminium cage for initial cameras disposition.

Commented [MM33]: Put a figure here

Commented [FM34R33]: Non ho trovato la figura di cui parlavamo, se la trovi puoi mandarmela?

Commented [BO35]: Your numbering has gone wrong somewhere

However, this technique did not prove effective due to the close proximity with the cameras. In fact, each camera comprises a ring of LED lights around the optics, which is seen as an object to mask, effectively reducing the capturing volume available. As a result, it was decided to proceed with a semicircle layout, as illustrated in **Figure 21**, with a radius of around 1.5m and all cameras facing towards the centre of the table.

Commented [BO36]: Don't understand this

3.4 3D printer objects and Power point instructions developed.

To correctly simulate each different grasp of the human hand, standardised objects with different shape and size were designed and 3D manufactured to simulate the industrial environment under consideration. The computer aided design software used was Solidworks (Dassault Systèmes, MA, US), while the 3D printer was a fused deposition modelling printer. The material selected for the objects



Figure 22. Projects in Solidworks (above) and printed objects (below)

Materials

was black polylactic acid (PLA), so to minimise any type of reflection and reduce the problem of mislabelling during the camera-based acquisition. Also, both tools (screwdriver and wrench) for the final task were 3D designed and printed, to avoid ferromagnetic interferences with the magnetometer inside the data glove. An overview of the objects used during the acquisition is given in **Figure 22** above.

In addition, as shown in **Figure 23**, a PowerPoint presentation showing the different grasps and tasks was created. This was used to instruct the participants during the data acquisition. This to improve task repeatability and speed of the process.

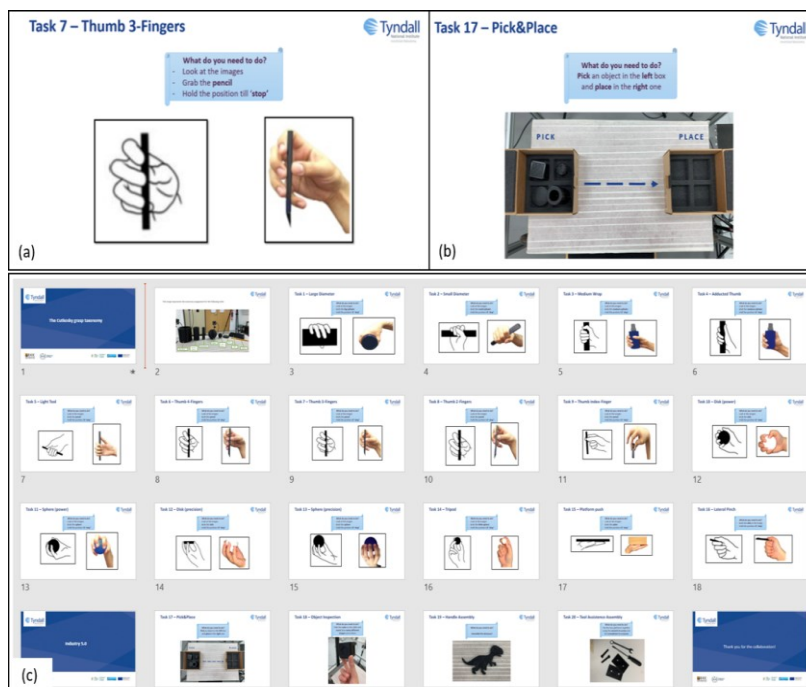


Figure 23. PowerPoint instructions created for the process. (a) example of Taxonomy's slide, (b) example of Industry's slide, (c) overall picture of the presentation.

4 Method

4.1 Database Creation

4.1.1 Data acquisition

Twenty subjects with ages ranging between 23 and 56 years old were recruited from the Tyndall National Institute. Before starting the data acquisition, each subject was briefed regarding the study and was given the opportunity to ask questions and to sign the required consent form under the terms of the ethics approval obtained to carry out the experiment with human subjects. Subsequently, the participant was asked to supply relevant demographic information, which is listed in **Table 4**, as well as to remove any metallic object (e.g., wristwatch, rings and earrings) that could cause reflections or could interfere with the magnetometers of the data glove sensors.

The data acquisition always commenced with the camera-based system, for which the subject was prepared by applying reflective markers as explained above (see Section 3.1). Once the entire set of defined tasks have been completed, a second round of activities is repeated, this time using the data glove as the data capture system. To communicate with and receive the data directly from the glove, a dedicated script in Matlab was created, which included the creation of a TCP/IP connection between the gloves integrated WiFi radio and the laptop and to carry out the data parsing. A second script¹ was created to generate joint angles from the quaternions transmitted from the glove, creating a .csv file for each data capture.

Table 4. Demographics information of participants

Subject	Hand	Sex	Age range	Subject	Hand	Sex	Age range
00	R	M	20 – 29	10	R	M	20 – 29
01	R	F	30 – 39	11	L	M	30 – 39
02	L	F	20 – 29	12	R	M	50 – 59
03	R	F	30 – 39	13	R	F	30 – 39
04	R	M	30 – 39	14	R	M	30 – 39
05	R	F	30 – 39	15	R	M	30 – 39
06	R	F	30 – 39	16	R	F	20 – 29

¹ View Appendix B.

07	R	M	30 – 39	17	R	M	30 – 39
08	R	M	20 – 29	18	R	F	20 – 29
09	R	M	30 – 39	19	R	M	30 – 39

For each MoCap technology the data acquisition was performed with the subject seated in front a small desk, which represent the centre of the capturing volume cameras as described previously. Each data acquisition includes analysis of 20 different tasks that each subject has to repeat three times; in **Figure 24** below is depicts an overview of the data acquisition flowchart.

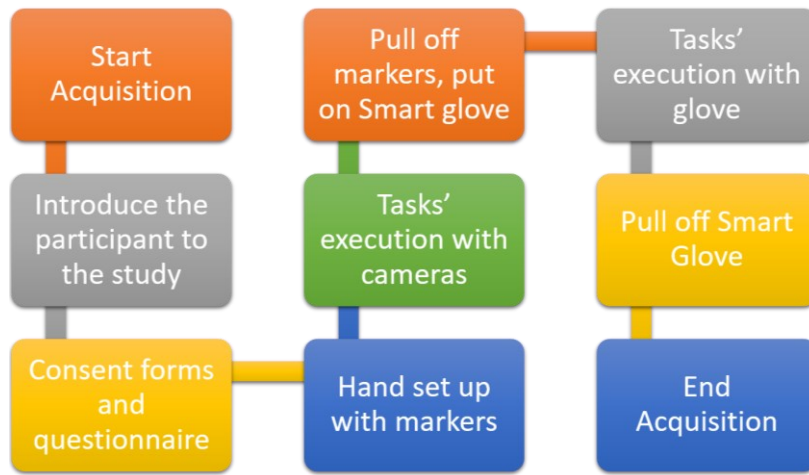


Figure 24. Workflow of the data acquisition process

4.1.2 Tasks explanation

Although the study focuses on the creation of a database of hand motion data to be used for task classification in industry for HRC-related applications, it was deemed appropriate to include a range of static hand grasps to foster a broader usage of the dataset.

4.1.2.1 Cutkosky Grasp Taxonomy

The Cutkosky Grasp Taxonomy is used to describe the different ways in which objects can be held by a hand, either human or robotic [60]. The categorization of grasps is based on the number of fingers used and their position relative to the object being grasped. It is useful for researchers and practitioners in fields such as

robotics, biomechanics, and human-robot interaction to understand the different types of grasps and their relative strengths and limitations. For example, certain grasps may be more stable or stronger than others, making them more appropriate for certain tasks. For this first part, therefore, we decided to include all 16 grasps identified by Cutkosky and to follow the order defined by him in principle shown in **Figure 25**.

This entire first section of the taxonomy was designed to be static for both technologies. Each task was recorded for a duration of at least 5 seconds and repeated three times. For each repetition the participant was asked to place the object on the table and pick it up again; this was to ensure high variability in the dataset as even if the grasp was made by the same subject, the grasping action can differ and hence guarantee more variability in the data set.

Commented [MM37]: Why?

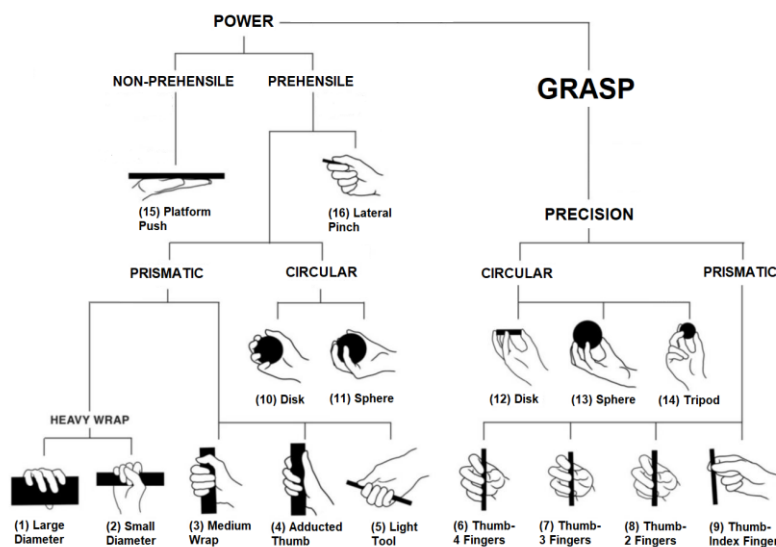


Figure 25. Cutkosky Grasp Taxonomy Task

4.1.2.2 Industry 4.0 Task

For the Industry 4.0 activities, four tasks that are classically performed in a HRC scenario have been simulated.

Method

- A. **Pick and place:** two boxes are placed at opposite sides on the table, the box on the left contains 4 objects and the box on the right is empty. The subject picks up each object one by one and place them on the box on the right side.
- B. **Object handover:** A 3D printed cube with geometrical shapes engraved on each side is placed on the table. The subject is asked to pick it up to count how many different shapes there are on the cube. This simulates the action of inspecting a component for quality assessment, typical in industrial production.
- C. **Hand assembly:** The subject is asked to put together the pieces of a jigsaw puzzle which has been disassembled and the pieces placed on the table in a random arrangement.
- D. **Tool assisted assembly:** The subject is asked to assemble two rectangular components with two bolts and two nuts. To do so, the subject uses a set of tools (screwdriver and wrench).

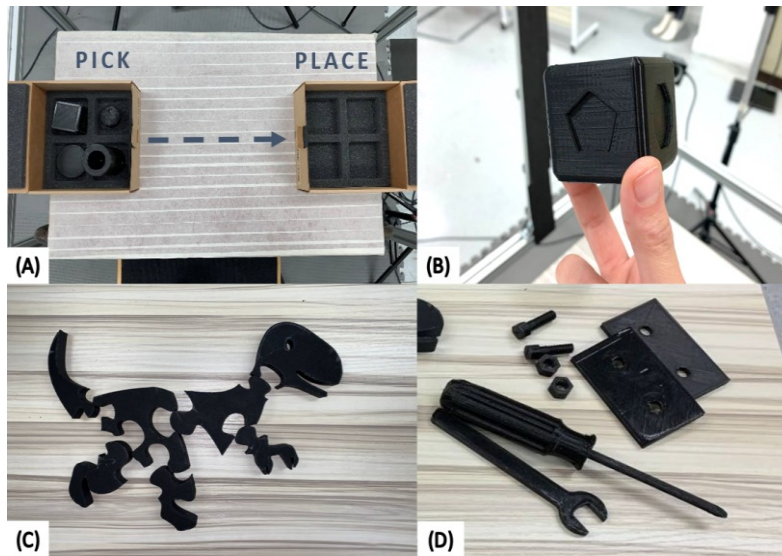


Figure 26. Industry tasks. (A) The two boxes used for pick-and-place. (B) The cubic object to observe during object inspection (C) A seven-piece puzzle to be assembled during the hand assembly task. (D) Assembly components and tools for the last task.

While for the acquisition of the static grasp configurations each acquisition time was set to approximately 30 seconds, for the industrial tasks no time limit was set.

Figure 26 shows the objects used to perform the four industrial activities.

In fact, the execution time to completion of such activities can vary from subject to subject. Participants were instructed to commence and complete each task by placing their hand in a resting position on the table, so to identify the start and the end of the task easier during the data post-processing.

Once again, the tasks were performed three times, with the exception of the last activity (Tool Assisted assembly), which was repeated four times to allow for at least two repetitions per mode (unscrewing/screwing).

4.1.3 Data processing

Two different tools were used for data processing depending on the technology, Motive was used for the camera-based method and Matlab was used for the data glove analytics.

4.1.3.1 Camera-based analytics

The data, in the case of the camera-based method, were processed with Motive software (Optitrack), the same software with which they were acquired (**Figure 27**). However, for both methods, the data cleaning followed the following steps:

1. *Data trimming*: for the taxonomy part, it was decided to consider only 240 frames (1 sec of acquisition) as, being a static part, they are sufficient to characterise the task. For the manufacturing part, on the other hand, the acquisition time was dictated by the time in which the participant performed the task; where the data was too corrupted by occlusions or artefacts, it was decided to cut the take so that the best 3600 frames (15 seconds of acquisition) remained.
2. *Data gap filling*: allows gaps resulting from camera occlusions to be filled using interpolation algorithms. In this case, a model-based algorithm was chosen to reconstruct gaps in a track by selecting a morphologically similar track. For example, in the case of a hand, if there was a gap in the track of

Commented [BO38]: Why not 6?

Commented [MM39]: Pizza spaghetti mandolino

Commented [FM40R39]: ahahahahaha ops

Method

the thumb marker, it is likely that the marker on the thumb's metacarpal bone had a similar signal, especially for small gaps. The pattern-based reconstruction algorithm took a signal indicated by the user as a reference and reconstructed the gap based on the selected signal. This allowed for precise and efficient reconstruction of gaps, improving the quality of the acquired data. Where not possible due to a lack of similar traces, cubic interpolation was used. Gap filling was only performed where the gap-size did not exceed 20 frames (83 ms) in the case of taxonomy and 100 frames in the case of industry (0.42 sec).

3. *Labelling*: although the model created by Motive is effective for tracking the hand, if movements are too fast or the markers disappear from the line of sight for few frames, the markers could be mislabelled. Therefore, each track was inspected, and markers relabelled manually if required.
4. *Smoothing*: a 4th-order Butterworth filter with cut-off frequency of 5Hz was applied to eliminate all fast oscillations and make the signal smoother on all takes.

Commented [MM41]: This is a bit counterintuitive. Why a static movement has a shorter gap allowance and a dynamic one a bigger one??

Commented [MM42]: It seems a bit low. Wasn't 10Hz?

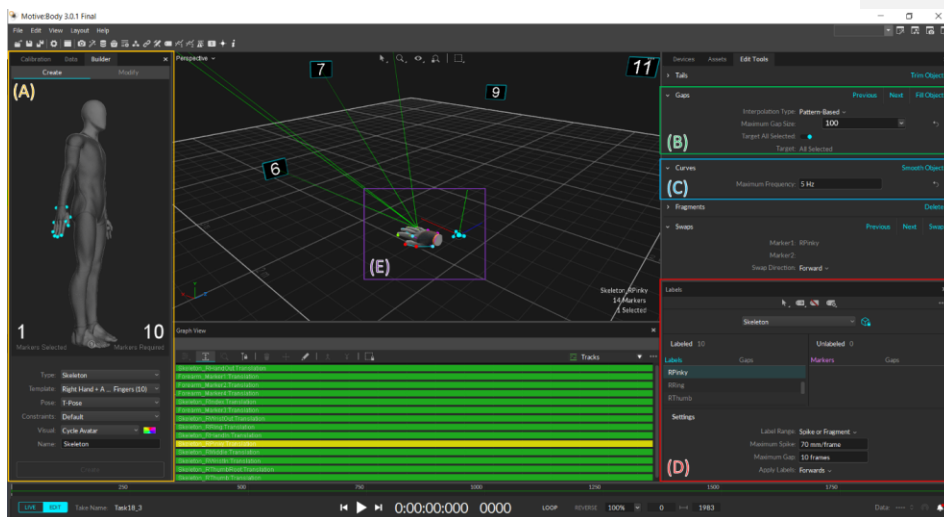


Figure 27. Motive Software interface. (A) Builder, where to create the hand model. (B) Gaps where to fill the gap. (C) Curves where to do the smoothing. (D) Labels where adjust the labels. (E) Hand model made with the Builder

4.1.3.2 Smart glove

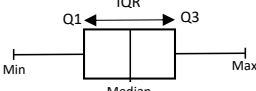
For the glove, the post-processing work was limited to applying a 4th-order Butterworth filter at the frequency @ 15Hz through a Matlab script (see Appendix A).

4.2 Features identification

From the acquired database a number of features, manually labelled according to the task being performed, were extracted and evaluated to train the different AI-based classifiers. Such features have been evaluated for both MoCap technologies, with few differences according to the typology of data available (pose data for the data glove and position data for camera). A Matlab script² was designed to automatically calculate the selected features for each subject and label them according to each industrial task.

Table 5 and **Table 6** describe the statistic and morphological features respectively, with the theoretical bases for clarity.

Table 5. Statistic features extract for each file.

Feature	Formulas	Description
Mean Value	$\bar{X} = \frac{\sum x_i}{N}$	Measure of central tendency that provides an estimate of the average value (\bar{X}) in set of data with N samples.
Standard Deviation	$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{X})^2}{N - 1}}$	Measure of the spread or variability of a data set with N samples. It is the square root of the variance and is used to indicate how much the data points in a data set deviate from the mean value (\bar{X}).
Median		Measure of central tendency that represents the middle value of a data set when the data points are arranged in order.
Interquartile Range (IQR)		Measure of variability that provides an estimate of the spread of a data set. It is calculated as the difference between 75 th (Q3) percentile and 25th percentile (Q1) of a data set.

² View Appendix B

Covariance	$\text{Cov}(X, Y) = \frac{\sum (x_i - \bar{X}) * (y_i - \bar{Y})}{N}$	Measure that provides information about the relationship between two variables. A positive covariance indicates that the two variables tend to increase or decrease together, while a negative one indicates that one variable tends to increase while the other decreases.
Entropy	$E = p(x) \log \frac{1}{p(x)}$	Measure of the uncertainty or randomness of a data set. The entropy of a data set is calculated as the sum of the probabilities of each event multiplied, $p(x)$, by the logarithm of the probability of the event. A higher entropy indicates that there is more uncertainty in the data set, while a lower entropy indicates that the data set is more predictable.
Shape Factor (SF)	$x_{SF} = \frac{x_{rms}}{\frac{1}{N} \sum_i^N x_i }$	Measure of the skewness of a data set. A positive SF indicates a positive skewness, where the data is more heavily weighted towards the right side of the distribution, while a negative SF indicates a negative skewness, where the data is more heavily weighted towards the left side of the distribution.

Table 6. Morphological feature calculated for each subject.

Feature	Description
Peak to Peak	Measure that provides the difference between the maximum and minimum values in a signal.
Find Peaks	It's a function that return the position of the peak in a specific window of observation, in this case we consider only the peaks that exceed 60% of the max peak.
No of Peaks	Count the number of peaks every window of observation.

4.2.1 Features Database creation.

The evaluation of the features was performed over 5-second time windows for both MoCap methods. Subsequently, the features were organized into tables to create two databases of sizes 745x173 and 833x137 for the glove and the camera-based method, respectively.

Each row of these databases corresponds to a 5-second window of the signal, while the columns contain the features calculated for that time window. The last column of each row contains the label of the reference task, necessary for the classification phase.

It is important to note that the features calculated for both technologies are the same, the difference in the column numbers for the data glove is due to the absence of the little and ring finger, and the consequent absence of the features related to them. This detail was taken into account when creating the feature database for the data glove, which is slightly smaller in size compared to the camera database.

4.3 AI classification

In the field of Artificial Intelligence, there is no unique and optimal solution, as solutions vary depending on the specifics of the final application. For this reason, we examined three different approaches to identify the most suitable solution for our specific case, taking into account the strengths and weaknesses of each approach.

4.3.1 Approach A: Basic Accuracy

Initially, the datasets were considered in their entirety, without dividing training sets and test sets. This was to get an idea of the basic accuracy of the datasets created.

Cross-validation is a technique that allows a model to be able to generalise, hence, to be able to classify well even a datum that does not belong to the training set. In particular, *k-fold cross validation* was used. This means dividing the dataset into k equal parts and training the model on $k-1$ parts by testing it on the remaining part.

After this process is repeated k times, the final performance is obtained by averaging the performance of each iteration.

This technique was applied to these datasets firstly to avoid overfitting, which occurs when a machine learning model is trained “too well” on the training data. Overfitting usually results in poor classification performances on new and unseen data. Secondly to better understand the performance of the tested predictive models. Specifically, for both datasets, a cross-validation with $k = 10$ was used, so with 20 subjects each fold consisted of 2 subjects. A visual example is given **Figure 28**.

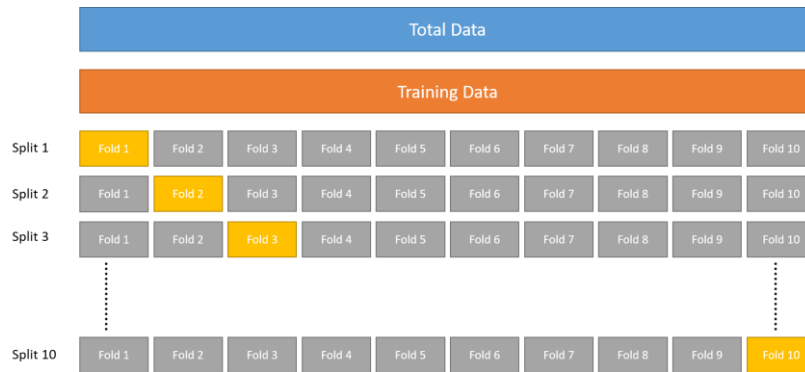


Figure 28. A scheme of 10-fold cross validation applied in Approach A, where the grey box is for training and the yellow one for validation.

Hyper-parameterization is the process of selecting the values of hyper-parameters, which are the parameters of a machine learning model that are not learned during training but are set a priori. Examples of hyper-parameters include the learning rate that determines the step size at each iteration while moving toward a minimum of a loss function, the number of hidden layers in a neural network, or the number of trees in a random forest.

The selection of hyper-parameters is a critical step in the machine learning process, as it can have a significant impact on the performance of the model. In this first approach, no hyper parameterisation was carried out in order to assess only the basic accuracy of the models tested.

Feature selection is the process of selecting a subset of relevant features from a larger set of features in a dataset, with the goal of improving the performance of a machine learning model. The selection of relevant features can be important for several reasons, e.g., improving the performance, reducing the complexity (less features can bring more interpretability) and/or the computational cost of the model. Among the most important feature selection algorithms are: Chi-square Test (CHI2), Analysis of Variance (ANOVA) or Maximum Relevance and Minimum Redundancy (MRMR). In this approach, no feature selection was performed in order to preserve basic accuracy.

Subsequently, models commonly used in the literature were implemented in the classification case. Specifically, the methods that were explored are SVM, Bagging Tree, and Wide Neural Network. Results will be shown in section 5.2.1

4.3.2 Approach B: Manual k-fold cross-validation

In the previous approach, the dataset was automatically divided into folds for the cross-validation process, but in this way, there was no control over the distribution of subjects in the different folds.

There are two left-handed people in the dataset, and although the left-handed/right-handed ratio in the dataset is consistent with the distribution in nature (1/10 [61]), it is not optimal in the classification case as they represent a small cohort. It was therefore of interest to us to proceed with a manual k-fold cross-validation to see if left-handedness had a negative influence on model performance.

A Matlab script was used to split the subjects into three folds. In this case, similar to the "classic" cross-validation approach, k iterations were performed, with one fold held back for validation in each iteration. In this way, the data was manually split for each iteration, with one part used for training and the other for validation. Then, to recreate the first split, folds 2 and 3 were combined and used for training while fold 1 was kept aside to validate the performance of the networks on the created training. This process was repeated for the next two permutations. For

clarity, a representative diagram of how the various groups were created is shown in **Figure 29**.

For both technologies, the following training method was adopted for each 'split':

- Step 1.** Training without feature selection;
- Step 2.** Training of the networks by performing feature selection of the best 18 features according to the ANOVA method;
- Step 3.** Training of the networks by performing feature selection of the best 18 features according to CHI2 method;
- Step 4.** Performance validation with the validation fold.

To select the networks on which to carry out the process described above, the Matlab Classification learner toolbox was used, thanks to which it was possible to obtain an overview of the networks that performed best on each of the technologies. For both, the three networks with the best performance were chosen. Specifically for the data recorded with the camera-based method, the methods of: Ensemble, SVM and Neural Network. While for the data recorded with the data glove: Ensemble, Trees, and Naive Bayes.

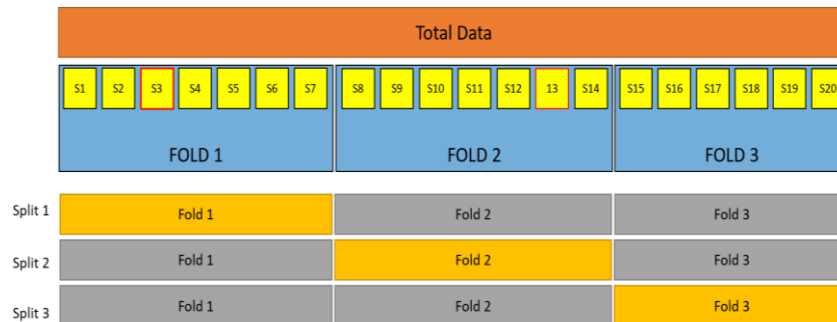


Figure 29. Manual 3-fold cross validation. The subjects framed in red are the left-handers. While the ochre rectangle is the validation fold and the grey the training ones.

Thus, at the end of each training procedure, performances were obtained, but just as in classic k-fold validation, averaging the results obtained in the various splits.

In this manual 3-fold cross-validation the final results (shown in section 5.2.2) were obtained by averaging the performances of each split.

4.3.3 Approach C: Balanced Dataset

In the two previous approaches, an overlooked limitation was the issue of imbalanced classes. In fact, some tasks have many more samples compared to others, as the time to completion differs. This problem was not taken into consideration previously, as the focus was on other limitations, such as the left-hand unbalancing. However, in the latest approach, we aimed to highlight the performance of the trained networks using a balanced dataset and automatic k-fold cross-validation. Since some classes have significantly more data than others, the resample function of the Pandas library in Python was used to address the class imbalance issue. This function works by randomly duplicating instances from the minority class or deleting instances from the majority class until the classes are balanced. By doing so, the classifier is less likely to be biased towards the majority class, which improves the overall accuracy of the model.

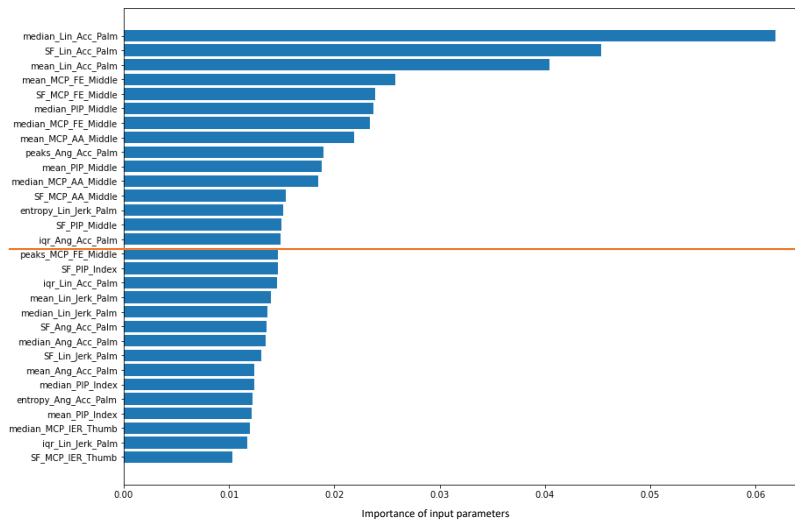


Figure 30. The 15 top features choose for Glove classification.

Prior to training, a feature selection process was conducted to identify the most important features for classification. The goal was to reduce the dimensionality of the datasets and only retain the most influential features for each task. The top 15

features were selected based on their ability to differentiate the tasks in both the data glove and motion capture databases (respectively **Figure 30** and **Figure 31**)

To optimize the performance of the classification networks, the hyper parameters of each network were optimized using the Classifier Learner toolbox in MATLAB. This toolbox allows for the optimization of a wide range of hyper parameters, including regularization, the number of hidden layers, and the number of neurons in each hidden layer. By optimizing these parameters, the networks are able to generalize better to unseen data, improving their overall performance.

Three different classification networks were trained on both databases, in particular: SVM, Random Forest and Neural Network. The performance of each network was evaluated using a 10-fold cross-validation, which helps to avoid overfitting and provides a more reliable estimate of the network's generalization performance. The networks were then tested on the remaining 20% of the data, which was held out during the training phase. The results of the study are reported in the section 0, and include a comparison of the performance of the classification networks on both the camera-based and data glove MoCap databases.

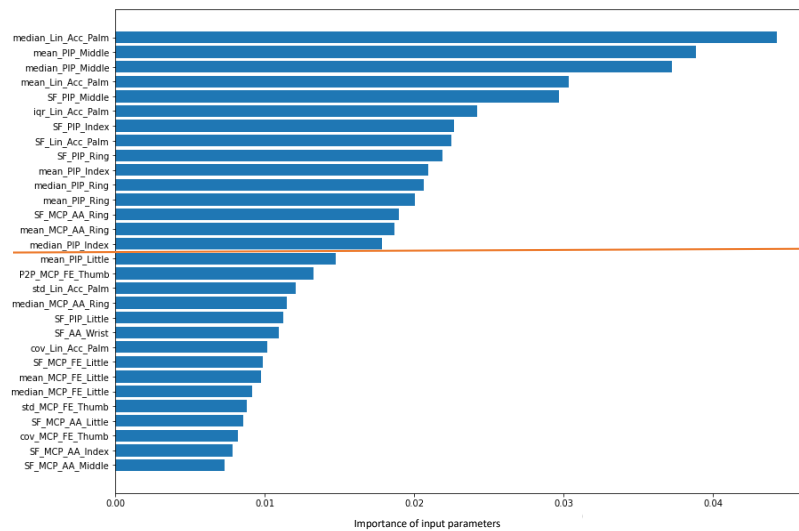


Figure 31. The 15 top features choose for Camera classification.

5 Results

5.1 Database

The creation of the **HANDMI4** (Motion capture database for Industry 4.0) dataset is the first significant outcome of this work [62]. The bulk of the database includes 5.63 GB of data, divided into folders according to the diagram shown in **Figure 32**. The initial macro division is based on the data collecting technique (Camera or Glove), followed by the static and dynamic tasks (Taxonomy or Industry tasks), and lastly the subdivision between the 20 subjects with the related .csv files within.

Commented [BO43]: Available at???

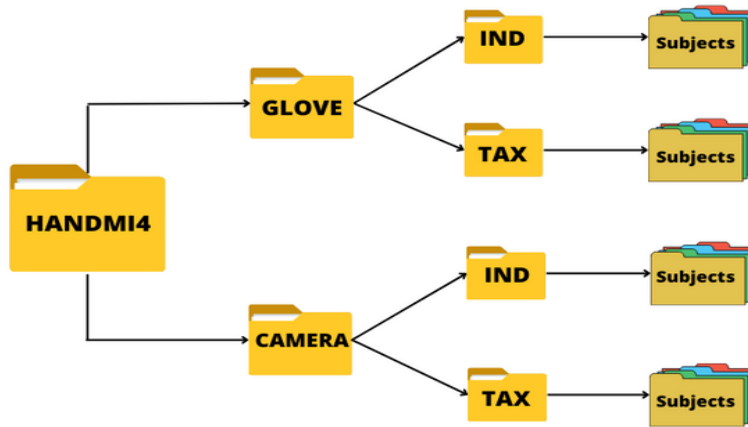


Figure 32. HANDMI4 partition

A label code³ was designed to improve navigation within the database's content, and as a result, the files have a unique name, making it easier to identify and process. Each label contains the following elements:

Table 7. File's labelling code

Label	Code	Description
ID	000	3-digit numeric participant identifier
Technology	GLV/OPT	3 letters which refer to the MoCap technology used
Task	TAX/IND	3 letters which refer to task performed
i-Task	00	2-digit numeric for the type of task
j-Repetition	00	2-digit numeric for the number of repetition (starting form 00)

³ Appendix C

Results

As an example, in the label *000GLVTAX0201*: 000 identifies the first participant, GLV stands for data glove, TAX refers to the Cutkosky's taxonomy, 02 refers to the second grasp of the Cutkosky's taxonomy, 01 means that this is the second repetition.

The content of the individual files, on the other hand, varies depending on the technology utilised (data glove or camera), this is because the two methods use distinct sensors and provide different types of results.

Indeed, in the case of the glove, the embedded IMUs provide information on accelerometers, magnetometers, and gyroscopes, from which the linear and angular motion may be calculated. Each .csv file contained in the glove section has the specific variables obtained through the acquisition on the first line. Additional information can be found in **Table 8**.

Table 8. Labelling code for .csv file in GLOVE, where Q = X, Y, Z; F= Index, Middle, Ring, Little, Thumb and ## equal to AA (Abduction/Adduction) or FE (Flexion/Extension)

Label code	Description
Prox	Proximal phalanx of the finger
Inter	Middle phalanx of the finger
Dist	Distal phalanx of the finger
Acc_Q	Acceleration along the component Q
Gyro_Q	Angular acceleration along the component Q
Mag_Q	Magnetometer along the component Q
Jerk_Acc_Q	Linear jerk along the component Q
Jerk_Gyro_Q	Angular jerk values along the component Q
PIP_F	Proximal Interphalangeal joint for the finger F
MCP_##_F	Metacarpophalangeal joint for the finger F in the range ##

In the case of cameras, on the other hand, the position in 3D space of each hand segment was provided. The camera system software also generates automatically the quaternions, which are a four-dimensional extension of complex numbers and are used to represent rotations and orientations in three-dimensional space. Each .csv file contained in the camera section has the variables obtained through the acquisition on the first line; Additional information can be found in **Table 9**.

Results

Table 9. Labelling code for .csv file in CAMERA, where # = 0, 1, 2, 3 (in the notation a quaternion q is defined as follow: $q = q_1 i + q_2 j + q_3 k + q_0$), $V = X, Y, Z$ and $F =$ Palm, Thumb, Index, Middle, Ring, Little, Forearm.

Label code	Description
Prox	Proximal phalanx of the finger
Inter	Middle phalanx of the finger
Dist	Distal phalanx of the finger
Pos_V	Position along the component V
F_q#	Quaternion # for the part F

The database also includes a text file providing instructions for handling and navigating the data. In the event of a partial download, similar metadata is included in the database's primary directories.

5.2 AI classification

For the sake of clarity, the results will be described according to the three approaches detailed in the methods section.

5.2.1 Approach A: Basic Accuracy

No hyper-parametrisation and feature selection was carried out on this first approach. The following tables show the performances of the three AI-classifier, for the camera (**Table 10**) and the glove (**Table 11**).

Table 10. Classification results for camera dataset.

Method	Accuracy [%]	Recall (macro avg)	Prediction speed [obs/sec]	Training time [sec]
SVM	90.9	89.2	~ 1500	7.316
Ensemble	87.1	84.0	~ 1300	35.27
Neural Net	89.2	87.2	~ 3600	30.61

Table 11. Classification results for glove dataset.

Method	Accuracy [%]	Recall (macro avg)	Prediction speed [obs/sec]	Training time [sec]
SVM	85.1	81.1	~ 8800	6.47
Ensemble	86.5	83.3	~1500	28.19
Neural Net	85.0	80.8	~17000	8.59

Results

Below in **Figure 33** and **Figure 34** the confusion matrix for each method is provided to offer a visual overview of the prediction rate of the classifiers.

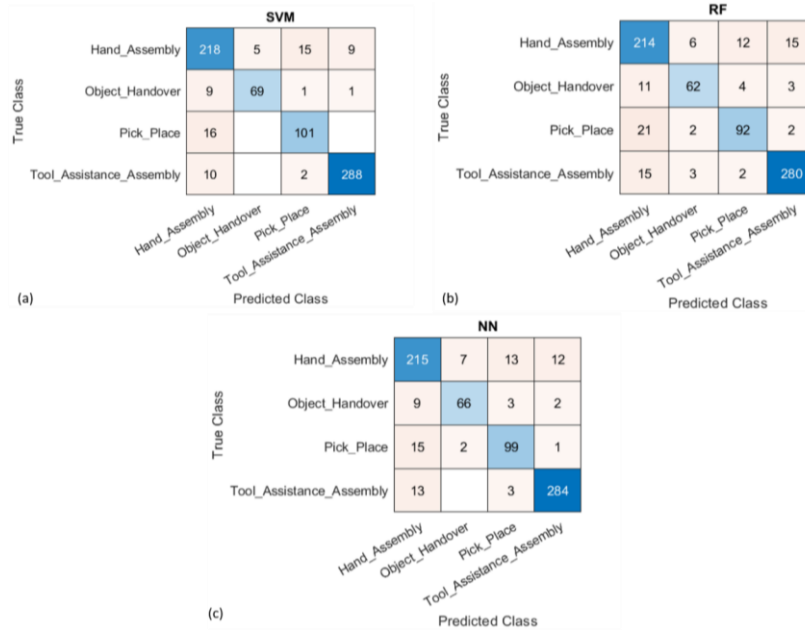


Figure 33. Confusion matrices for the chosen network in the camera dataset

5.2.2 Approach B: Manual k-fold cross-validation

As outlined in Section 4.3.2, a manual 3-fold cross-validation approach was employed to assess the impact of the two left-handed variables in our dataset. The tables below present the average accuracy scores for both technologies, with three different scenarios: no feature selection, feature selection with CHI2 algorithm, and feature selection using the ANOVA algorithm.

Table 12. Classification results for camera dataset without feature selection.

Method	Accuracy_val [%]	Accuracy_test [%]	Prediction speed [obs/sec]	Training time [sec]
SVM	93.97	75.80	1967	90.92
Ensemble	87.03	72.47	607	197.09
NN	92.87	78.40	3300	126.70

Results

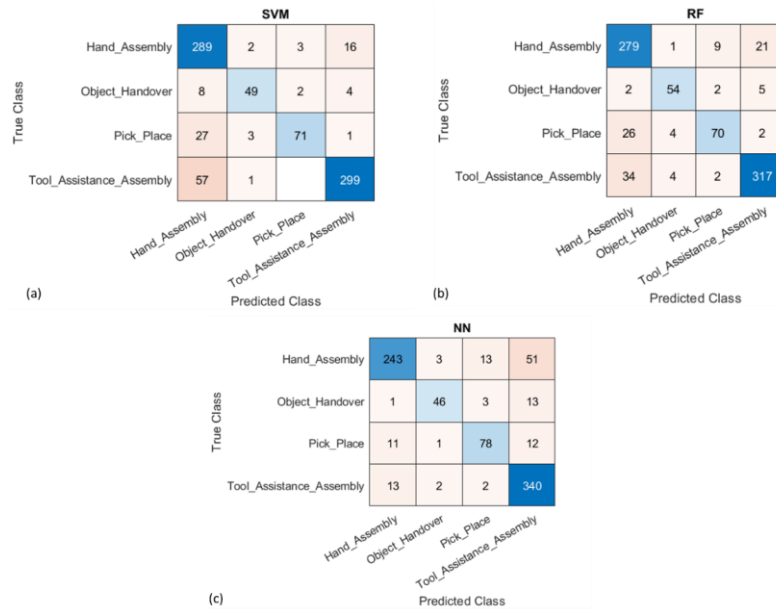


Figure 34. Confusion matrices for the chosen network in the glove dataset

Table 13. Classification results for camera dataset with 18/173 features selected with CHI2.

Method	Accuracy_val [%]	Accuracy_test [%]	Prediction speed [obs/sec]	Training time [sec]
SVM	87.33	68.23	3733	148.04
Ensemble	85.93	69.40	881	182.39
NN	83.17	69.83	11633	207.99

Table 14. Classification results for camera dataset with 18/173 features selected with ANOVA.

Method	Accuracy_val [%]	Accuracy_test [%]	Prediction speed [obs/sec]	Training time [sec]
SVM	88.50	63.83	4400	65.51
Ensemble	85.13	71.53	147	151.02
NN	80.80	66.67	8700	126.52

Results

Table 15. Classification results for glove dataset without feature selection.

Method	Accuracy_val [%]	Accuracy_test [%]	Prediction speed [obs/sec]	Training time [sec]
SVM	89.70	77.87	1203	172.03
Ensemble	71.90	71.33	1589	126.96
NN	80.07	67.97	5767	30.28

Table 16. Classification results for glove dataset with 18/173 features selected with CHI2.

Method	Accuracy_val [%]	Accuracy_test [%]	Prediction speed [obs/sec]	Training time [sec]
SVM	88.43	75.80	633	110.52
Ensemble	81.23	72.77	1753	79.80
NN	78.90	66.13	13000	54.35

Table 17. Classification results for glove dataset with 18/173 features selected with ANOVA.

Method	Accuracy_val [%]	Accuracy_test [%]	Prediction speed [obs/sec]	Training time [sec]
SVM	87.20	77.87	600	114.38
Ensemble	83.00	76.23	683	76.65
NN	80.13	72.13	6007	54.10

5.2.3 Approach C: Balanced Dataset

In this section, we present the results obtained showing the dataset balanced with respect to the majority class. For each method, we selected the top 15 features that had the greatest impact in the classification. We used k=10 cross-validation, setting aside 20% of the total dataset for testing. Additionally, we provide the confusion matrices for each selected test and validation method.

Table 18. Classification results for glove dataset.

Method	Accuracy val [%]	Accuracy test [%]	Recall_test (macro avg)	Prediction speed [obs/sec]	Training time [sec]
SVM	96.5	93.7	93.7	5800	1417.0
Ensemble	95.5	95.7	95.6	2700	297.02
NN	97.2	94.5	94.4	6800	1108.3

Results

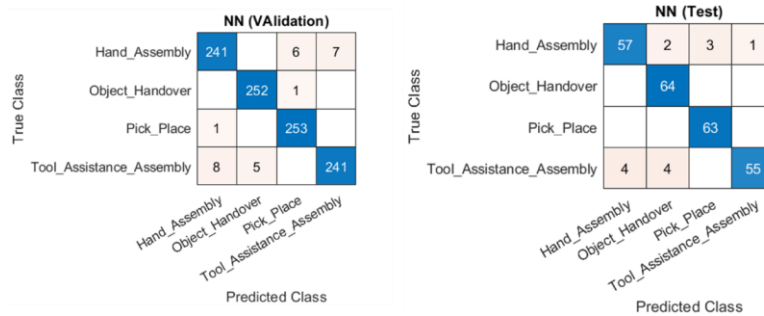


Figure 35. Confusion matrix of validation and test for the Neural Network net.

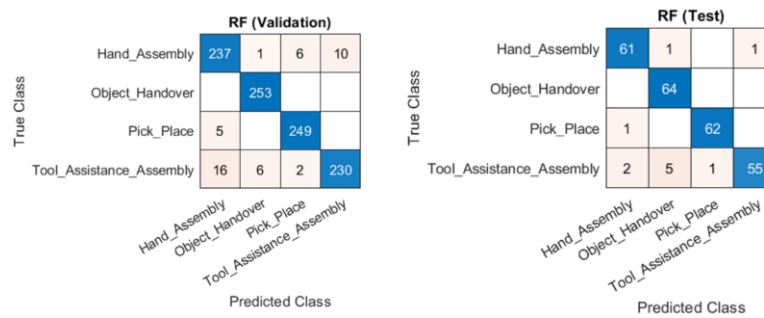


Figure 36. Confusion matrix of validation and test for the Random Forest net.

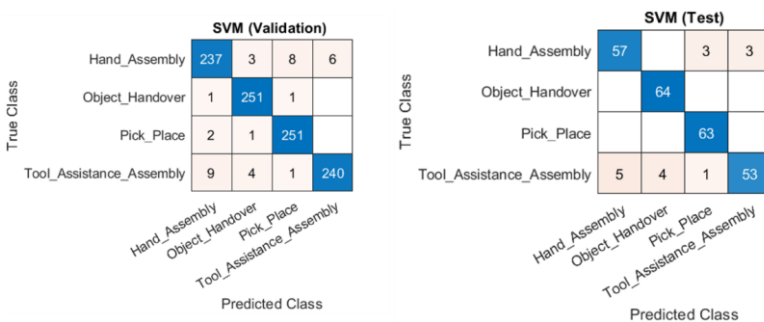


Figure 37. Confusion matrix of validation and test for the SVM net.

The tables and figures that follows below are for the camera dataset.

Results

Table 19. Classification results for camera dataset.

Method	Accuracy val [%]	Accuracy test [%]	Recall test (macro avg)	Prediction speed [obs/sec]	Training time [sec]
SVM	94.8	92.1	92.0	4900	735.01
Ensemble	93.9	93.8	93.7	2800	361.19
NN	95.3	95.0	95.0	9200	1227.5

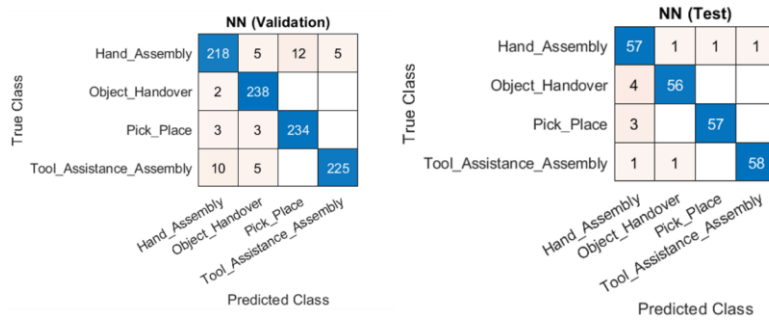


Figure 39. Confusion matrix of validation and test for the Neural Network net.

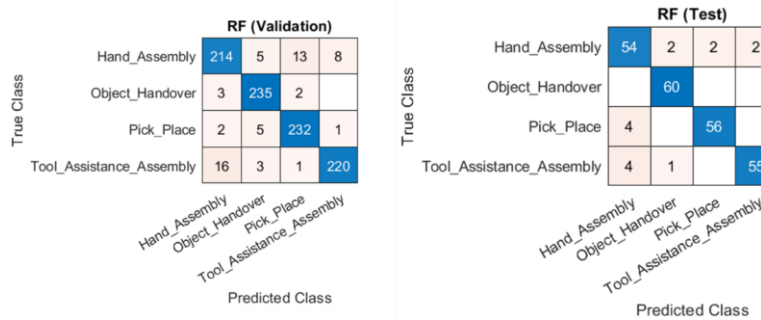


Figure 40. Confusion matrix of validation and test for the Random Forest net.

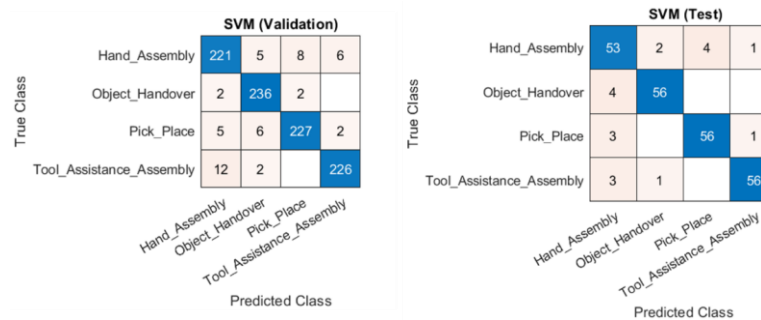


Figure 38. Confusion matrix of validation and test for the SVM net

Results

This final table below include all the accuracy of the various approach to have an overall vision.

Table 20. The table shows the overall accuracy [%] of the different approach. Specifically, B1 refers to the second approach without feature selection, B2 18 features selected with CHI2, and B3 18 features selected with ANOVA.

Approach	Technology	SVM	RF	NN
A	Camera	90.9	87.1	89.2
	Glove	85.1	86.5	85.0
B1	Camera	75.8	72.5	78.4
	Glove	77.9	71.3	68.0
B2	Camera	68.2	69.4	69.8
	Glove	75.8	72.8	66.1
B3	Camera	63.8	71.5	66.7
	Glove	77.9	76.2	72.1
C	Camera	92.1	93.8	95.0
	Glove	93.7	95.7	94.5

6 Discussion

6.1 HANDMI4 Dataset

In terms of the number of subjects and data quantity, the hand motion capture dataset created during this thesis work is well positioned with respect to similar datasets developed in the field (see **Table 2**). Moreover, our dataset presents a more usable and comprehensive set of data, including data collected from two different MoCap technologies (cameras and data glove) and, in addition to industry-related dynamic tasks, the inclusion of the full taxonomy of static grasps, enabling its usage in many fields of application.

The grasp taxonomy aspect of the database can be used, for example, for biomechanical and/or anatomical studies, to refine and improve the design of handed tools or wearable devices. As for the industry and manufacturing task aspects of the database, such data represent a singularity in the field as, at the moment, the majority of the datasets in the literature focus on daily hand gestures, like open a bottle or drink a glass of water, so movement that are not useful in an industry environment and as a consequence in a study that want to focus on industry task.

In an industrial context, the use of wearable technologies, such as gloves with embedded sensors, can enable teleoperation of robots just by tracking the hand gestures and interpreting relevant motions and gestures. This can offer a huge advantage in those scenarios where robots need to rapidly change and adapt their defined assembly operations, to manufacture either customised products or change their tasks within modular production lines to optimise efficiencies according to batch size under production. In fact, with data gloves, human co-workers can control the autonomous behaviour of the robot and take full control of the robot if needed. This simplifies the human-machine interaction and can help in improving the efficiency and productivity of the collaborative work performed. Thanks to rapid developments in the field of wearable sensing technologies and low latency communication (e.g., LoRa system and 5G), it is plausible to foresee a

Commented [B044]: Explain the relevance and impact of this better

future in which the use of these devices will increase, and become even more integrated into the industrial environment, enabling improvements in safety, quality, productivity and worker wellbeing.

Although the true power of HANDMI4 lies in the hidden layers of information that can be extracted from the motion data, such as the importance of tracking certain part of the hand, as it has been demonstrated in this work, some criticalities have been identified during and after the data acquisition. These include:

- *Different duration of the acquisition time for the different tasks.* Although some tasks require more time to be performed, having the exact same amount of frames per tasks could ensure a more balanced database for AI processing. In its current form, the database is currently unbalanced (more data frames for specific tasks) which impacts on the classification performance, as highlighted in the result session.
- *Unbalanced number of left-handed subjects vs right-handed ones.* Despite the fact that the ratio in the database is consistent with the natural distribution of left-handed population (1/10) in the general population, if the number of left-handed are not present in a significant number they can affect significantly on the training process, or can be labelled as outliers, effectively decreasing the quantity and variability of the data in general.
- *Presence of not-a-number (NaN) in the database.* During the database creation phase, we have decided to not alter the original raw data produced by the different MoCap devices. These however can produce NaN every time there is an obstruction on a marker, particularly in the case of the camera-based MoCap data acquisition system. Once again, this has an impact in the overall balance of the database.

Overall, HANDMI4 represents a powerful resource for researchers working on AI-based task classification and biomechanics. By making HANDMI4 openly available and providing explicit instructions on how the data were collected and organised,

Commented [MM45]: Use bullet points when you can. They help to break down complex concept more easily for the readers. They also help to visualise better interconnections between the different items.

other academics could be encouraged to use, implement, and expand the dataset which is freely available to the research community [62].

Commented [BO46]: Where?

6.2 AI Classifier

In this work, we implemented several AI-based classification strategies, as each technique have pros and cons that can largely have more or less impact in term of performances according to the final application (e.g., safety, robot autonomy, etc.). To ensure this consideration in the accuracy assessment of each classification strategies, we thoroughly analysed our dataset to identify its strengths and limitations before delving into each approach.

Commented [BO47]: In terms of what?

Commented [BO48]: Rephrase this - I don't really understand what you mean

Prior to carrying out such an analysis, it is important to highlight that the features used for the classification were extracted using the same methodology for each signal, regardless of the method used. We selected a 5-second window to evaluate our time series features, to ensure adequate task-related variation of the data. This could per se represent a limitation. In fact, some of the recorded tasks have a duration varying between 15-20 seconds, reducing the number of features produced to 3-4 per acquisition. Moreover, such an approach might be not suitable for task classification in real time applications. Reducing the window in this way could reduce the time required to produce a classification, but could also decrease the variability of the data, hence their task-related characteristics. Another approach might involve the use of a sliding window, which consider past events in the evaluation of the feature. For instance, a 5-second window could consider 1 second of new data and 4 seconds of old data. This could produce features every second and ensure enough variation of the morphology of the data to be captured by the feature evaluation, particularly important for peaks-related features.

Besides windowing, the types of extracted features also played a role in classification. This study focused on statistical features and features related to signal morphology, such as peak's height and number of peaks (see section 4.2). However, it is important to note that other types of features, such as those related to time-

frequency representations or wavelet analysis, could potentially improve classification accuracy. Therefore, exploring a wider range of features and different space domain would be a valuable direction for further research.

6.2.1 Approach A: Basic Accuracy

This first approach aimed to evaluate at a very basic level the overall consistency and effectiveness of the database as a training tool for the different classification methods, keeping hyper-parameter tuning and feature selection out of the process. By using the k-fold cross-validation technique with $k=10$, we were able to train and validate the models on multiple subsets of the data, which provided a reliable assessment of the classification performance for each method.

One limitation of this approach was the lack of control in the distribution of left-handed subjects within the subdivisions of the dataset during the k-fold validation, which could have affected the accuracy value. Therefore, in subsequent approaches, we ensured that the training and testing sets were properly balanced to account for the presence of left-handed individuals.

Despite this limitation, our analysis of the results (section 5.2.1) showed that the best-performing classification technique varied depending on the data acquisition method. In the case of the camera-based approach, the SVM algorithm achieved the highest accuracy, while in the case of the glove-based approach, a random forest algorithm performed best.

Moreover, we observed that the confusion matrix for the glove-based approach had a higher misclassification rate compared to the camera-based approach. This could be attributed to the fact that the signals were acquired only for three fingers (thumb, index, and middle) in the glove-based approach, while the camera-based approach provided information about the whole hand. As a result, the glove-based approach may have been more susceptible to misclassification due to the limited input information.

Overall, this initial approach provided us with valuable insights into the behaviour of different classification methods and highlighted the strengths and weaknesses

Commented [MM49]: Wait, we used only index, middle and thumb finger for the generation of the features for both techniques, right? So this cannot be the reason.

Commented [MF50R49]: No, it's like is written. We use index, middle and thumb for the glove and the whole hand for the camera

of the different approaches. With this knowledge, we were able to refine our training in subsequent analyses, leading to improved classification accuracy. Furthermore, the results of this first step demonstrated adequacy of the structure and quality of the dataset, which, with further analysis and refinement of the training process, could lead to excellent results in the field of human-robot collaboration.

6.2.2 Approach B: Manual 3-Fold Cross Validation

In this approach, a k-fold cross-validation was manually performed to ensure an adequate distribution of left-handed subjects between training and testing subsets. The results obtained in section 5.2.2 showed a test accuracy that was approximately 10 percentage points lower than the validation accuracy. This discrepancy may have been due to **overfitting** of the training model as described previously. To mitigate for overfitting, various techniques were investigated to avoid overfitting, including **regularization** when a penalty is added to the loss function, **dropout** that randomly drops out neurons during training to prevent overfitting in deep learning models, and **early stopping** to stop training a model when the performance on a validation set starts to degrade, could be applied in future works. It is also important to note that the reported results were the final ones of the cross-validation process, i.e., the ones averaged over the three iterations. The individual results from the various process steps showed lower accuracy when both left-handed subjects were in the training set but were in line with the accuracy of the previous approach when the two subjects were divided between training and testing data sets. However, even when the two subjects are divided into training and testing, it does not represent an ideal condition for the classifier, as it is forced to learn from just a single example, which is highly not robust.

Furthermore, in this approach, the impact of feature selection on the results was also tested. To apply this method in industry, it is crucial to understand where the information lies in both technologies to refine acquisition methods with fewer sensors, making it more portable, wearable, and cost effective for the different industrial applications. As seen in the results, the highest accuracies were obtained when all features were selected, which was not surprising since the classifier had

Commented [MM51]: Have you explain somewhere what overfitting is. If yes, please ignore this comment. Otherwise just give a very short description.

Commented [MM52]: Here again just give a bit more explanation about these terms.

more information to infer the output. However, eliminating approximately 90% of the features and input information resulted in only a few percentage points lost in accuracy. This is particularly significant for the test set, indicating the substantial impact of just few critical features (e.g., linear and angular palm acceleration, MCP of the middle, PIP of the index, and linear jerk of the palm). This should be further investigated in future works by trying classifiers with different features and features selections.

Moreover, the results showed that, for both technologies, the 18 features selected using the ANOVA method performed better than those selected using the CHI2 method, demonstrating ANOVA as the more efficient feature selection method in our case. However, it is important to note that the optimal feature selection method may vary depending on the dataset, especially on the type of data, and further investigation is needed to determine the best method for other datasets.

In conclusion, this approach showed promising results in classifying left and right-handed movements using two different technologies. However, there is still room for improvement, particularly in mitigating overfitting and balancing the dataset. Further investigation is also needed to determine the optimal feature selection method and improve the method's generalizability to other datasets and industries.

6.2.3 Approach C: Balanced Dataset

In the final approach to the data analysis, the aim was to address the limitations of the previous approaches which have been used for classification. One major issue identified was the imbalance in the dataset, where some classes had significantly fewer instances compared to others. This imbalance can cause a biased model towards the more populous classes and result in poor performance in classifying the minority classes. To overcome this problem, the dataset was balanced by adding instances to the minor classes to make them equal in size to the most populous class, as described in detail in section 4.3.3.

Commented [MM53]: Which ones? It's important here to remind the reader.

Commented [BO54R53]: agreed

Commented [MM55]: Depending on what in the dataset?

Commented [MM56]: Adding instances taken randomly or obtained with a specific algorithm? Please, specify properly here because it's a crucial point

Commented [MF57R56]: I've already explain it in the method, i've also to explain here?

Commented [BO58R56]: A reminder is always good IMO

Commented [BO59]: Blah blah (reminder, As described in detail in section XXX

Discussion

The results of this approach demonstrated that a balanced dataset improved the performance of the classification model. This improvement was particularly evident in the confusion matrix, which showed a significant reduction in the number of misclassified instances.

In addition to balancing the dataset, this final approach investigated also used a different feature selection method compared to the previous approaches. ANOVA and CHI2 are two commonly used feature selection methods that select features based on their correlation with the target variable. These methods eliminate highly correlated features, which can lead to a reduction in the number of features used in the model. However, in some cases, highly correlated features can contain complementary information that is useful for classification.

Therefore, in this approach, instead of using ANOVA or CHI2, the top 15 features with the highest weight in the classification were selected thanks to a preliminary training that allowed to see which features weighted more on the classification. This approach ensured that the most relevant features for classification were included, regardless of their correlation with the target variable. Furthermore, this approach also aimed to minimize the number of features used in the model, which is beneficial for the system's efficiency.

Overall, this latest approach addressed the limitations of the previous approaches and yielded better results in classifying the dataset in term of the overall accuracy. Balancing the dataset and using a different feature selection method improved the performance of the classification model, and selecting the top features ensured that the most relevant information was included while minimizing the system's complexity.

6.2.4 Summary analysis of Results

Overall, these results, regardless of the approach used, demonstrated the enormous potential of the HANDMI4 dataset. This is particularly significant within the field of AI, where there are numerous approaches to tackling any particular prob-

Commented [MM60]: Obtained HOW? Very crucial point here!

Commented [BO61]: In terms of X and Y

Discussion

lem. The approaches outlined in this work are just a small selection of the possibilities available. One of the primary strengths of the HANDMI4 dataset is its versatility, which allows it to be used for a variety of applications, such as biomechanical studies, fatigue, AI classification or ergonomic studies. Additionally, the high level of accuracy for the different approaches in this study indicates that there may be other techniques that could be applied to the dataset to yield even better results. The potential for developing devices incorporating edge-AI (e.g., tiny-ML) which can support industrial activities with minimal impact on the worker is very significant. By reducing the sensing components required, such a system could make significant contributions to enhancing efficiency, reducing costs, and minimizing environmental impact of any manufacturing process. However, developing such a system requires significant effort and prototyping, and there are still many challenges and unsolved research problems in this space that must be addressed (e.g., ergonomics, usability). In light of these challenges, continued research in this area is essential.

Commented [BO62]: Such as

7 Conclusions and future works

The aim of this thesis was to investigate the use of motion capture technologies for hand motion tracking and test a potential usage as industry task classification in the context of collaborative robotics in industry 4.0. To this end, a comprehensive hand motion-capture dataset, called HANDMI4, incorporating a wide taxonomy of hand grasp activity examples and a variety of classic industry-related task activities, was created. This database is now available for download on an open repository [62], as is the data acquisition protocol developed to enable future research in this topic. As a use case for such database, a series of AI-based classifiers were trained and a selected group of statistical and morphological features extracted from the database itself. Data collection and classification performance suggested a potential benefit in using MoCap technology to improve the level of autonomy of collaborative robots to make them more proactive and safer in conjunction with the tasks being performed by the operator.

The primary achievement of this work, the creation and dissemination of HANDMI4, represents a powerful research tool in the field of Industry 4.0 and other application domains. Moreover, the open accessibility and the dissemination of the data acquisition protocol could enable other researchers to increase the amount of data of HANDMI4, effectively improving its value in a variety of AI applications. The current cohort of 20 subjects well positions the database in terms of the amount of data in comparison with similar datasets in the field. One potential area for dataset augmentation is in respect of the limited variability of subject-related characteristics. For instance, including more left handed subjects in the dataset, including a significant number of subjects across a variety of age ranges, and incorporating subjects with a broader range of manual-labour skills could massively improve the robustness of the dataset as an AI-algorithms training tool. This could also open up new avenues of research or attract other potential usage of such database, such as in biomechanical modelling or the design of biomimetic end-effectors.

Commented [B063]: link

Conclusions and future works

As for the data acquisition protocol, we have also identified the reduced number of cameras as one of the reasons for the limitation in the tracking capability of the MoCap camera-system. Increasing the number of cameras and distributing the location of the cameras all around the periphery of the data capture volume could improve the tracking performance. The implementation of multi modal data acquisition solutions might be necessary in order to ensure an adequate level of tracking, especially for those tasks where the hand naturally folds or where objects being manipulated obstructs the markers. To this end, the use of smaller markers or active markers, or the manufacturing of light transparent objects (e.g., Plexiglas bench and tools) could offer a solution to the issue of such obstructions. Although this would confine the data acquisition to an in lab environment, as the aforementioned solutions are hardly implementable in a real life manufacturing scenario. Additionally, smaller markers can increase wearability of the measurement systems used and consequently the ergonomics for the subject can be improved from a useability and human factors perspective. This might also enable the tracking of each single hand segment (e.g., distal, intermediate, and proximal phalange), eliminating the need for software based hand models, which represent only an estimation of the actual hand pose, with obvious limitation in pose accuracy.

As for the wearable device measurement system being used, the hardware design of the data glove was found to have a critical role in ensuring an adequate adhesion of the sensors to the hand's segment, as well as in its wearability. In the current design, these two elements are in opposition. The open design of the Tyndall smart glove (hardware parts fastened to the hand via thin Velcro straps rather than mounted on a glove) has in fact the advantage of improving breathability of the hand skin and washability of the device but could reduce its adhesion to the fingers, introducing motion artefacts which may impact on system accuracy. Different data glove designs could be tested in future iterations to address these issues. For instance, a potential solution might be to use a textile glove that meets the fit criteria described above and also integrates the necessary sensors within it.

Commented [MM64]: You need to explain here or in the discussion why the use of hand model is a problem!! You are giving for granted that people know what you are talking about.

Here you need to explain that hand models generate just an estimation of the hand configuration based on the markers position and the biomechanical model of the hand they use. So they are not ideal for very accurate hand motion tracking.

Commented [MM65]: Not necessarily

Commented [MM66]: For example? You need to provide some speculative solutions or it would be just too vague. You have worked with this devices, you have work them so how better then you can offer some suggestions.

Conclusions and future works

The second main achievement of this work was the application of HANDMI4 for AI-based task classification. In particular, the resulted identification of the features that play an important role in the classification and where this information can be extracted more effectively from the MoCap system. In fact, the results have shown that for both technologies, the majority of useful information for classification is contained within the middle finger and palm. This implies that a 12-IMU system or a 12-camera system, such as the one used in this study, may not be necessary. Instead, a simplified system incorporating just one or two sensors placed on these anatomical parts could be sufficient. This would be a significant breakthrough in the field of HRC due to the portability and deployability of such a system. By using fewer sensors, the overall system could be made more portable, cost-effective, and user-friendly, which would enable a broader range of applications. Moreover, using fewer sensors could simplify the data processing and classification tasks, reducing the complexity of the system and improving its accuracy. However, it should be noted that while using fewer sensors may be beneficial, it could also result in a loss of some useful information. Thus, it is important to carefully consider the trade-off between the number of sensors and the amount of useful information that can be obtained from them.

Another area of focus for future work in this space might be the integration of the different MoCap and IMU based systems used in this work with other technologies such as virtual reality, which can create new possibilities for applications in fields such as gaming, teleoperation, and rehabilitation.

Overall, the future work should endeavour to create a system that is more versatile, efficient, and user-friendly, opening up new avenues for research and applications in hand movements' analysis.

Commented [MM67]: Based on the result section, give more insights on how this can be used to implement a minimal wearable device that implement maybe just 1 IMU to classify tasks in industrial context.

Commented [MM68]: What do you mean with the system? The AI classifier, the MoCap + classifier, other?

Commented [MM69]: Here again is not clear what you are referring to. Are you talking of a potential minimal glove which implement also classification on the edge?

References

- [1] "Contemplas, professional motion analysis software." <https://contemplas.com/en/motion-analysis/3d-analysis/> (accessed 22/02/2023).
- [2] P. Inc. "the Impulse X2 glove. ." <http://www.phasespace.com/gloves.html> (accessed 25/12/2022).
- [3] Q. f. S. Analysis. "Passive markers." <https://www.quinticsports.com/accessories/> (accessed 4/11/2022).
- [4] U. o. O. Kristian Nymoen. "Infrared marker-based motion capture." <https://www.futurelearn.com/info/courses/music-moves/0/steps/12692> (accessed 26/10/22).
- [5] M. Bianchi, P. Salaris, A. Turco, N. Carbonaro, and A. Bicchi, "On the use of postural synergies to improve human hand pose reconstruction," presented at the 2012 IEEE Haptics Symposium (HAPTICS), 2012.
- [6] Y. Park, J. Lee, and J. Bae, "Development of a finger motion measurement system using linear potentiometers," presented at the 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2014.
- [7] I. Tian, N. Thalmann, D. Thalmann, and J. Zheng, "The Making of a 3D-Printed, Cable-Driven, Single-Model, Lightweight Humanoid Robotic Hand," *Frontiers in Robotics and AI*, vol. 4, 12/04 2017, doi: 10.3389/frobt.2017.00065.
- [8] S. Cobos, M. Ferre, M. A. Sanchez Uran, J. Ortego, and C. Pena, "Efficient human hand kinematics for manipulation tasks," presented at the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.
- [9] Y. Ma, Z.-H. Mao, W. Jia, C. Li, J. Yang, and M. Sun, "Magnetic Hand Tracking for Human-Computer Interface," *IEEE Transactions on Magnetism*, vol. 47, no. 5, pp. 970-973, 2011, doi: 10.1109/tmag.2010.2076401.
- [10] V. K. Nanayakkara, G. Cotugno, N. Vitzilaios, D. Venetsanos, T. Nanayakkara, and M. N. Sahinkaya, "The Role of Morphology of the Thumb in Anthropomorphic Grasping: A Review," *Frontiers in Mechanical Engineering*, vol. 3, 2017, doi: 10.3389/fmech.2017.00005.
- [11] T. L. Baldi, S. Scheggi, L. Meli, M. Mohammadi, and D. Prattichizzo, "GESTO: A Glove for Enhanced Sensing and Touching Based on Inertial and Magnetic Sensors for Hand Tracking and Cutaneous Feedback," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 1066-1076, 2017, doi: 10.1109/thms.2017.2720667.
- [12] B. Grieco *et al.*, "A Low-g 3 Axis Accelerometer for Emerging Automotive Applications," 2004, pp. 211-222.
- [13] M. Caeiro-Rodriguez, I. Otero-Gonzalez, F. A. Mikic-Fonte, and M. Llamas-Nistal, "A Systematic Review of Commercial Smart Gloves: Current Status and Applications," *Sensors (Basel)*, vol. 21, no. 8, Apr 10 2021, doi: 10.3390/s21082667.
- [14] A. Gustus, G. Stillfried, J. Visser, H. Jorntell, and P. van der Smagt, "Human hand modelling: kinematics, dynamics, applications," *Biol Cybern*, vol. 106, no. 11-12, pp. 741-55, Dec 2012, doi: 10.1007/s00422-012-0532-4.
- [15] N. Saadon-Grosman, Y. Loewenstein, and S. Arzy, "The 'creatures' of the human cortical somatosensory system," *Brain Commun*, vol. 2, no. 1, p. fcaa003, 2020, doi: 10.1093/braincomms/fcaa003.

References

- [16] G. D. Pasquale, "Glove-based systems for medical applications: review of recent advancements," *Journal of Textile Engineering & Fashion Technology*, vol. 4, no. 3, 2018, doi: 10.15406/jteft.2018.04.00153.
- [17] E. Britannica. "Hand anatomy." <https://www.britannica.com/science/hand-anatomy> (accessed 23 September 2022).
- [18] C. S. LLC. cyberglovesystems.com (accessed 29 September, 2022).
- [19] Ł. Jaworski and R. Karpiński, "Biomechanics of the Human Hand," *Journal of Technology and Exploitation in Mechanical Engineering*, vol. 3, no. 1, pp. 28-33, 2017, doi: 10.35784/jtme.536.
- [20] M. Santello, M. Flanders, and J. F. Soechting, "Postural Hand Synergies for Tool Use," *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10105-10115, 1998, doi: 10.1523/jneurosci.18-23-10105.1998.
- [21] V. T. Minh, N. Katushin, and J. Pumwa, "Motion tracking glove for augmented reality and virtual reality," *Paladyn, Journal of Behavioral Robotics*, vol. 10, no. 1, pp. 160-166, 2019, doi: 10.1515/pjbr-2019-0012.
- [22] M. Field, D. Stirling, F. Naghdy, and Z. Pan, "Motion capture in robotics review," presented at the 2009 IEEE International Conference on Control and Automation, 2009.
- [23] M. Rahul, "Review on Motion Capture Technology," *Global journal of computer science and technology*, 2018.
- [24] L. Wanqing, Z. Zhengyou, and L. Zicheng, "Expandable Data-Driven Graphical Modeling of Human Actions Based on Salient Postures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1499-1510, 2008, doi: 10.1109/tcsvt.2008.2005597.
- [25] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion," *International Journal of Computer Vision*, vol. 87, no. 1, p. 4, 2009/08/05 2009, doi: 10.1007/s11263-009-0273-6.
- [26] F. S. Parizi, E. Whitmire, and S. Patel, "AuraRing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1-28, 2019, doi: 10.1145/3369831.
- [27] K. Mitobe *et al.*, "Development of a motion capture system for a hand using a magnetic three dimensional position sensor," presented at the ACM SIGGRAPH 2006 Research posters on - SIGGRAPH '06, 2006.
- [28] I. Adochiei, S. Patlagica, D. I. Voiculescu, M. Stanculescu, O. Drosu, and V. Vita, "Virtual Reality Assisted Hand Rehabilitation through the Use of a Smart Glove," presented at the 2019 E-Health and Bioengineering Conference (EHB), 2019.
- [29] C. Mizera, T. Delrieu, V. Weistroffer, C. Andriot, A. Decatoire, and J. P. Gazeau, "Evaluation of Hand-Tracking Systems in Teleoperation and Virtual Dexterous Manipulation," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1642-1655, 2020, doi: 10.1109/jsen.2019.2947612.
- [30] M. A. Diftler, C. J. Culbert, R. O. Ambrose, R. Platt, and W. J. Bluethmann, "Evolution of the NASA/DARPA Robonaut control system," presented at the 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), 2003.
- [31] S. Ghate, L. Yu, K. Du, C. T. Lim, and J. C. Yeo, "Sensorized fabric glove as game controller for rehabilitation," presented at the 2020 IEEE Sensors, 2020.
- [32] S. Zhang *et al.*, "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances," *Sensors (Basel)*, vol. 22, no. 4, Feb 14 2022, doi: 10.3390/s22041476.

References

- [33] A. Kristoffersson and M. Linden, "A Systematic Review of Wearable Sensors for Monitoring Physical Activity," *Sensors (Basel)*, vol. 22, no. 2, Jan 12 2022, doi: 10.3390/s22020573.
- [34] T. D. Cordeiro *et al.*, "Open Source Cloud Computing Platforms," presented at the 2010 Ninth International Conference on Grid and Cloud Computing, 2010.
- [35] M. Weston, "Wearable surveillance – a step too far?," *Strategic HR Review*, vol. 14, no. 6, pp. 214-219, 2015, doi: 10.1108/shr-09-2015-0072.
- [36] F. Niemann *et al.*, "LARA: Creating a Dataset for Human Activity Recognition in Logistics Using Semantic Attributes," *Sensors (Basel)*, vol. 20, no. 15, Jul 22 2020, doi: 10.3390/s20154083.
- [37] D. Z. Roggen, P. HCI gestures. [Online]. Available: <http://har-dataset.org/doku.php?id=wiki:dataset>
- [38] K. Forster, D. Roggen, and G. Troster, "Unsupervised Classifier Self-Calibration through Repeated Context Occurrences: Is there Robustness against Sensor Displacement to Gain?," presented at the 2009 International Symposium on Wearable Computers, 2009.
- [39] C. T. Mandery, O.; Do, M.; Vahrenkamp, N.; Asfour, T. *KIT Whole-Body Human Motion Database*. [Online]. Available: <http://motion-database.humanoids.kit.edu/>
- [40] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "The KIT whole-body human motion database," presented at the 2015 International Conference on Advanced Robotics (ICAR), 2015.
- [41] D. C. Roggen, A.; Nguyen-Dinh, L.V.; Chavarriaga, R.; Sagha, H.; Digumarti, S.T. *Activity Recognition Challenge/Opportunity*. [Online]. Available: <http://www.opportunity-project.eu/challenge.html>
- [42] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," presented at the 2010 Seventh International Conference on Networked Sensing Systems (INSS), 2010.
- [43] A. B. Bulling, U.; Schiele, B. *Andreas-Bulling/ActRecTut*. [Online]. Available: <http://github.com/andreas-bulling/ActRecTut>
- [44] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, no. 3, pp. 1-33, 2014, doi: 10.1145/2499621.
- [45] P. Zappi *et al.*, "Activity Recognition from On-Body Sensors: Accuracy-Power Trade-Off by Dynamic Sensor Selection," in *Wireless Sensor Networks*, Berlin, Heidelberg, R. Verdone, Ed., 2008// 2008: Springer Berlin Heidelberg, pp. 17-33.
- [46] R. C. Jafari, C.; Kehtarnavaz, N. *UTD Multimodal Human Action Dataset (UTD-MHAD)*. [Online]. Available: <http://personal.utdallas.edu/~kehtar/UTD-MHAD.html>
- [47] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *2015 IEEE International Conference on Image Processing (ICIP)*, 27-30 Sept. 2015 2015, pp. 168-172, doi: 10.1109/ICIP.2015.7350781.
- [48] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," presented at the 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE), 2017.
- [49] L. Chen, S. Wang, W. Fan, J. Sun, and S. Naoi, "Beyond human recognition: A CNN-based framework for handwritten character recognition," presented at the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015.

References

- [50] R. Mutegeki and D. S. Han, "A CNN-LSTM Approach to Human Activity Recognition," presented at the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), 2020.
- [51] C. Hou, "A study on IMU-Based Human Activity Recognition Using Deep Learning and Traditional Machine Learning," presented at the 2020 5th International Conference on Computer and Communication Systems (ICCCS), 2020.
- [52] Z. Yao, Z. Wang, W. Liu, Y. Liu, and J. Pan, "Speech emotion recognition using fusion of three multi-task learning-based classifiers: HSF-DNN, MS-CNN and LLD-RNN," *Speech Communication*, vol. 120, pp. 11-19, 2020, doi: 10.1016/j.specom.2020.03.005.
- [53] Y. Fan, X. Lu, D. Li, and Y. Liu, "Video-based emotion recognition using CNN-RNN and C3D hybrid networks," presented at the Proceedings of the 18th ACM International Conference on Multimodal Interaction, 2016.
- [54] D. Phan, N. Nguyen, P. N. Pathirana, M. Horne, L. Power, and D. Szmulewicz, "A Random Forest Approach for Quantifying Gait Ataxia With Truncal and Peripheral Measurements Using Multiple Wearable Sensors," *IEEE Sensors Journal*, vol. 20, no. 2, pp. 723-734, 2020, doi: 10.1109/jsen.2019.2943879.
- [55] N. Halim, "Stochastic recognition of human daily activities via hybrid descriptors and random forest using wearable sensors," *Array*, vol. 15, 2022, doi: 10.1016/j.array.2022.100190.
- [56] OptiTrack. "Camera Models." <https://www.optitrack.com/cameras/compare/?products=602&600&602&603&604> (accessed 02/02/23).
- [57] OptiTrack. "Motive by Optitrack." <https://optitrack.com/software/motive/> (accessed 10/11/22, 2022).
- [58] M. P. Wilk, J. Torres-Sanchez, S. Tedesco, and B. O'Flynn, "Wearable Human Computer Interface for Control Within Immersive VAMR Gaming Environments Using Data Glove and Hand Gestures," presented at the 2018 IEEE Games, Entertainment, Media Conference (GEM), 2018.
- [59] A. J. DUPREE. "How long does it take to manufacture a hardware product?" <http://www.andrewdupree.com/blog/2016/8/3/how-long-does-it-take-to-manufacture-a-hardware-product> (accessed 10/11/22, 2022).
- [60] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 269-279, 1989.
- [61] C. Hardyck and L. F. Petrinovich, "Left-handedness," *Psychological Bulletin*, vol. 84, no. 3, pp. 385-404, 1977, doi: 10.1037/0033-2909.84.3.385.
- [62] F. Mongelli and M. Menolotto. *HANDMI4 (HAND Motion capture data for Industry 4.0)*, doi: <https://doi.org/10.21227/c6t8-ge47>.

Appendix A

```
% Glove data analysis: joints evaluation
% Tyndall National Institute
% 19/09/2022 Francesca Mongelli, Matteo Menolotto

clc
close all
clear all

% read the data from the .mat file
subject = 'sub25\'; % puts the value of the subject that want to analysed
fileList = dir(strcat('Data_Collection\Glove\',subject));
for rec=3:size(fileList,1)
    Glove = load(strcat('Data_Collection\Glove\',subject, fileList(rec).name));

% Evaluate Jerk
Acc = Glove.Acc;
Gyro = Glove.Gyro;
Mag = Glove.Mag;
for i=1:size(Acc,2)
    Jerk_Acc(:,i) = diff(Acc(:,i));
    Jerk_Gyro(:,i) = diff(Gyro(:,i));
end
Jerk_Acc(size(Acc,1),:) = NaN;
Jerk_Gyro(size(Gyro,1),:) = NaN;

% Get Glove quaternions and assign to the correct hand part
Quat = Glove.Quaternion;
Quat_Index_Prox = quaternion(Quat(:,9:12));
Quat_Index_Inter = quaternion(Quat(:,13:16));
Quat_Middle_Prox = quaternion(Quat(:,33:36));
Quat_Middle_Inter = quaternion(Quat(:,37:40));
Quat_Ring_Prox = quaternion(Quat(:,21:24));
Quat_Ring_Inter = quaternion(Quat(:,25:28));
Quat_Little_Prox = quaternion(Quat(45:48));
Quat_Little_Inter = quaternion(Quat(:,49:52));
Quat_Thumb_Prox = quaternion(Quat(:,5:8));
Quat_Thumb_Inter = quaternion(Quat(:,57:60));
Quat_Forearm = quaternion(Quat(:,65:68));
Quat_Palm = quaternion(Quat(:,1:4));

%% PIPs Flexion-Extension
% PIP extraction
PIP_Index_quat = conj(Quat_Index_Prox).*Quat_Index_Inter;
PIP_Index_eul = eulerd(PIP_Index_quat,'ZYX','frame');
PIP_Index = PIP_Index_eul(:,3);
% address the 180 deg over-wrap
for i=1:numel(PIP_Index)
    if PIP_Index(i)<-120
        PIP_Index(i) = PIP_Index(i)+360;
    end
end
PIP_Middle_quat = conj(Quat_Middle_Prox).*Quat_Middle_Inter;
PIP_Middle_eul = eulerd(PIP_Middle_quat,'ZYX','frame');
PIP_Middle = PIP_Middle_eul(:,3);
% address the 180 deg over-wrap
for i=1:numel(PIP_Middle)
    if PIP_Middle(i)<-120
        PIP_Middle(i) = PIP_Middle(i)+360;
    end
end
```

Appendix A

```

    end
end
PIP_Ring_quat = conj(Quat_Ring_Prox).*Quat_Ring_Inter;
PIP_Ring_eul = eulerd(PIP_Ring_quat,'ZYX','frame');
PIP_Ring = PIP_Ring_eul(:,3);
% address the 180 deg over-wrap
for i=1:numel(PIP_Ring)
    if PIP_Ring(i)<-120
        PIP_Ring(i) = PIP_Ring(i)+360;
    end
end
PIP_Little_quat = conj(Quat_Little_Prox).*Quat_Little_Inter;
PIP_Little_eul = eulerd(PIP_Little_quat,'ZYX','frame');
PIP_Little = PIP_Little_eul(:,3);
% address the 180 deg over-wrap
for i=1:numel(PIP_Little)
    if PIP_Little(i)<-120
        PIP_Little(i) = PIP_Little(i)+360;
    end
end
PIP_Thumb_quat = conj(Quat_Thumb_Prox).*Quat_Thumb_Inter;
PIP_Thumb_eul = eulerd(PIP_Thumb_quat,'ZYX','frame');
PIP_Thumb = PIP_Thumb_eul(:,3);
% address the 180 deg over-wrap
for i=1:numel(PIP_Thumb)
    if PIP_Thumb(i)<-120
        PIP_Thumb(i) = PIP_Thumb(i)+360;
    end
end

% smoothing (Uncomment if angle jerk evaluation is needed)
PIP_Index = medfilt1(PIP_Index,100);
PIP_Middle = medfilt1(PIP_Middle,100);
PIP_Ring = medfilt1(PIP_Ring,100);
PIP_Little = medfilt1(PIP_Little,100);
PIP_Thumb = medfilt1(PIP_Thumb,100);

%% MCPs Flexion-Extension
MCP_Index_quat = conj(Quat_Palm).*Quat_Index_Prox;
MCP_Index_eul = eulerd(MCP_Index_quat,'ZYX','frame');
MCP_FE_Index = MCP_Index_eul(:,3);
MCP_Middle_quat = conj(Quat_Palm).*Quat_Middle_Prox;
MCP_Middle_eul = eulerd(MCP_Middle_quat,'ZYX','frame');
MCP_FE_Middle = MCP_Middle_eul(:,3);
MCP_Ring_quat = conj(Quat_Palm).*Quat_Ring_Prox;
MCP_Ring_eul = eulerd(MCP_Ring_quat,'ZYX','frame');
MCP_FE_Ring = MCP_Ring_eul(:,3);
MCP_Little_quat = conj(Quat_Palm).*Quat_Little_Prox;
MCP_Little_eul = eulerd(MCP_Little_quat,'ZYX','frame');
MCP_FE_Little = MCP_Little_eul(:,3);
MCP_Thumb_quat = conj(Quat_Palm).*Quat_Thumb_Prox;
MCP_Thumb_eul = eulerd(MCP_Thumb_quat,'ZYX','frame');
MCP_FE_Thumb = MCP_Thumb_eul(:,3);

% smoothing (Uncomment if angle jerk evaluation is needed)
MCP_FE_Index = medfilt1(MCP_FE_Index,100);
MCP_FE_Middle = medfilt1(MCP_FE_Middle,100);
MCP_FE_Ring = medfilt1(MCP_FE_Ring,100);
MCP_FE_Little = medfilt1(MCP_FE_Little,100);
MCP_FE_Thumb = medfilt1(MCP_FE_Thumb,100);

%% MCPs Abduction-Adduction
MCP_AA_Index = MCP_Index_eul(:,1);

```

Appendix A

```

MCP_AA_Middle = MCP_Middle_eul(:,1);
MCP_AA_Ring = MCP_Ring_eul(:,1);
MCP_AA_Little = MCP_Little_eul(:,1);
MCP_AA_Thumb = MCP_Thumb_eul(:,1);

% smoothing (Uncomment if angle jerk evaluation is needed)
MCP_AA_Index = medfilt1(MCP_AA_Index,100);
MCP_AA_Middle = medfilt1(MCP_AA_Middle,100);
MCP_AA_Ring = medfilt1(MCP_AA_Ring,100);
MCP_AA_Little = medfilt1(MCP_AA_Little,100);
MCP_AA_Thumb = medfilt1(MCP_AA_Thumb,100);
%

%% Thumb CMC Internal-External Rotation
MCP_IER_Thumb= MCP_Thumb_eul(:,2);

% smoothing (Uncomment if angle jerk evaluation is needed)
MCP_IER_Thumb = medfilt1(MCP_IER_Thumb,100);

%% Wrist Flexion-Extension & Abduction-Adduction
Wrist_quat = conj(Quat_Forearm).*Quat_Palm;
Wrist_eul = eulerd(MCP_Index_quat,'ZYX','frame');
Wrist_FE = Wrist_eul(:,3);
Wrist_AA= Wrist_eul(:,1);

% smoothing (Uncomment if angle jerk evaluation is needed)
Wrist_FE = medfilt1(Wrist_FE,100);
Wrist_AA = medfilt1(Wrist_AA,100);

%% Creation of the csv file
%data_time = datetime('now','TimeZone','local','Format','yyyyMMdd_hhmmss'); %
get date
T = table(Acc, Gyro, Mag, Jerk_Acc, Jerk_Gyro, ...
    PIP_Index, PIP_Middle, PIP_Ring, PIP_Little, PIP_Thumb, ...
    MCP_FE_Index, MCP_FE_Middle, MCP_FE_Ring, MCP_FE_Little, MCP_FE_Thumb, ...
    MCP_AA_Index, MCP_AA_Middle, MCP_AA_Ring, MCP_AA_Little, MCP_AA_Thumb, ...
    MCP_IER_Thumb, Wrist_FE, Wrist_AA);
filepath = strcat('Data_Classification\Database\',subject); % here the filepath
of the new folder
[status,msg] = mkdir(filepath); % create the folder
for a specific subject
filename = erase(fileList(rec).name, ".mat"); % delete from the
filename the extension .mat
writetable(T,filepath,sprintf('Glove_data_%s.csv',filename)), 'Writ-
eRowNames',true)

clear Acc Gyro Mag Jerk_Acc Jerk_Gyro PIP_Index PIP_Middle PIP_Ring ...
    PIP_Little PIP_Thumb MCP_FE_Index MCP_FE_Middle MCP_FE_Ring ...
    MCP_FE_Little MCP_FE_Thumb MCP_AA_Index MCP_AA_Middle MCP_AA_Ring ...
    MCP_AA_Little MCP_AA_Thumb MCP_IER_Thumb Wrist_FE Wrist_AA ...

end

```

Appendix B

```

%% Task Identification Using Hand Tracking Technology
% Francesca Mongelli, Matteo Menolotto - Tyndall National Institute
% 10/01/2023
%%
clc
clear all
close all

%% Constants declaration
Tasks = ["Pick_Place" "Object_Handover" "Hand_Assembly" "Tool_Assistance_Assembly"];
SampTime = 190; % Sample time
sec = 5; % 5 seconds time window (950 frames)
windowWidth = SampTime*sec; % 5 seconds time window (950 frames)
Time_wind = windowWidth/SampTime;
varNames = {'mean_PIP_Thumb', 'mean_PIP_Index', 'mean_PIP_Middle',...
            'mean_MCP_FE_Thumb', 'mean_MCP_FE_Index', 'mean_MCP_FE_Middle',...
            'mean_MCP_AA_Thumb', 'mean_MCP_AA_Index', 'mean_MCP_AA_Middle',...
            'mean_MCP_IER_Thumb', 'mean_FE_Wrist', 'mean_AA_Wrist', 'mean_Lin_Acc_Palm',...
            'mean_Ang_Acc_Palm', 'mean_Lin_Jerk_Palm', 'mean_Ang_Jerk_Palm',...
            'std_PIP_Thumb', 'std_PIP_Index', 'std_PIP_Middle',...
            'std_MCP_FE_Thumb', 'std_MCP_FE_Index', 'std_MCP_FE_Middle',...
            'std_MCP_AA_Thumb', 'std_MCP_AA_Index', 'std_MCP_AA_Middle',...
            'std_MCP_IER_Thumb', 'std_FE_Wrist', 'std_AA_Wrist', 'std_Lin_Acc_Palm',...
            'std_Ang_Acc_Palm', 'std_Lin_Jerk_Palm', 'std_Ang_Jerk_Palm',...
            'median_PIP_Thumb', 'median_PIP_Index', 'median_PIP_Middle',...
            'median_MCP_FE_Thumb', 'median_MCP_FE_Index', 'median_MCP_FE_Middle',...
            'median_MCP_AA_Thumb', 'median_MCP_AA_Index', 'median_MCP_AA_Middle',...
            'median_MCP_IER_Thumb', 'median_FE_Wrist', 'median_AA_Wrist', 'median_Lin_Acc_Palm',...
            'median_Ang_Acc_Palm', 'median_Lin_Jerk_Palm', 'median_Ang_Jerk_Palm',...
            'iqr_PIP_Thumb', 'iqr_PIP_Index', 'iqr_PIP_Middle',...
            'iqr_MCP_FE_Thumb', 'iqr_MCP_FE_Index', 'iqr_MCP_FE_Middle',...
            'iqr_MCP_AA_Thumb', 'iqr_MCP_AA_Index', 'iqr_MCP_AA_Middle',...
            'iqr_MCP_IER_Thumb', 'iqr_FE_Wrist', 'iqr_AA_Wrist', 'iqr_Lin_Acc_Palm',...
            'iqr_Ang_Acc_Palm', 'iqr_Lin_Jerk_Palm', 'iqr_Ang_Jerk_Palm',...
            'cov_PIP_Thumb', 'cov_PIP_Index', 'cov_PIP_Middle',...
            'cov_MCP_FE_Thumb', 'cov_MCP_FE_Index', 'cov_MCP_FE_Middle',...
            'cov_MCP_AA_Thumb', 'cov_MCP_AA_Index', 'cov_MCP_AA_Middle',...
            'cov_MCP_IER_Thumb', 'cov_FE_Wrist', 'cov_AA_Wrist', 'cov_Lin_Acc_Palm',...
            'cov_Ang_Acc_Palm', 'cov_Lin_Jerk_Palm', 'cov_Ang_Jerk_Palm',...
            'entropy_PIP_Thumb', 'entropy_PIP_Index', 'entropy_PIP_Middle',...
            'entropy_MCP_FE_Thumb', 'entropy_MCP_FE_Index', 'entropy_MCP_FE_Middle',...
            'entropy_MCP_AA_Thumb', 'entropy_MCP_AA_Index', 'entropy_MCP_AA_Middle',...
            'entropy_MCP_IER_Thumb', 'entropy_FE_Wrist', 'entropy_AA_Wrist',...
            'entropy_Lin_Acc_Palm', 'entropy_Ang_Acc_Palm', 'entropy_Lin_Jerk_Palm',...
            'SF_PIP_Thumb', 'SF_PIP_Index', 'SF_PIP_Middle',...
            'SF_MCP_FE_Thumb', 'SF_MCP_FE_Index', 'SF_MCP_FE_Middle',...
            'SF_MCP_AA_Thumb', 'SF_MCP_AA_Index', 'SF_MCP_AA_Middle',...
            'SF_MCP_IER_Thumb', 'SF_FE_Wrist', 'SF_AA_Wrist', 'SF_Lin_Acc_Palm',...
            'SF_Ang_Acc_Palm', 'SF_Lin_Jerk_Palm', 'SF_Ang_Jerk_Palm',...
            'P2P_PIP_Thumb', 'P2P_PIP_Index', 'P2P_PIP_Middle',...
            'P2P_MCP_FE_Thumb', 'P2P_MCP_FE_Index', 'P2P_MCP_FE_Middle',...
            'P2P_MCP_AA_Thumb', 'P2P_MCP_AA_Index', 'P2P_MCP_AA_Middle',...
            'P2P_MCP_IER_Thumb', 'P2P_FE_Wrist', 'P2P_AA_Wrist', 'P2P_Lin_Acc_Palm',...
            'P2P_Ang_Acc_Palm', 'P2P_Lin_Jerk_Palm', 'P2P_Ang_Jerk_Palm',...
            'peaks_PIP_Thumb', 'peaks_PIP_Index', 'peaks_PIP_Middle',...
            'peaks_MCP_FE_Thumb', 'peaks_MCP_FE_Index', 'peaks_MCP_FE_Middle',...

```



```

'peaks_FE_Wrist', 'peaks_Lin_Acc_Palm', 'peaks_Ang_Acc_Palm', 'Task_name'};

%% Import all the tasks (csv) of a subject
for subjects = 1:20
    Task_list = dir(strcat(['\\FS1\Docs4\francesca.mongelli\My Documents
\Database\Database_Features_Prep\GLOVE\sub'], num2str(subjects), '\*.csv'));
    for file=1:numel(Task_list)
        Task = readtable(strcat('Database_Features_Prep\GLOVE\sub', num2str(sub-
jects), '\', Task_list(file).name));
        Data_Acquisition{file} = Task;
    end

    %% Feature Extraction (Joint angles)
    for file=1:numel(Task_list)
        time = 0:1/SampTime:(size(Data_Acquisition{file},1)-1)/SampTime;
        for i=0:floor(size(Data_Acquisition{file},1)/windowWidth)-1
            % MEAN VALUE
            mean_PIP_Thumb(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,1}, 'omitnan');
            mean_PIP_Index(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,2}, 'omitnan');
            mean_PIP_Middle(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,3}, 'omitnan');
            mean_MCP_FE_Thumb(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,6}, 'omitnan');
            mean_MCP_FE_Index(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,7}, 'omitnan');
            mean_MCP_FE_Middle(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,8}, 'omitnan');
            mean_MCP_AA_Thumb(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,11}, 'omitnan');
            mean_MCP_AA_Index(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,12}, 'omitnan');
            mean_MCP_AA_Middle(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,13}, 'omitnan');
            mean_MCP_IER_Thumb(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,16}, 'omitnan');
            mean_FE_Wrist(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,17}, 'omitnan');
            mean_AA_Wrist(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,18}, 'omitnan');
            mean_Ang_Acc_Palm(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,19}, 'omitnan');
            mean_Lin_Acc_Palm(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,20}, 'omitnan');
            mean_Lin_Jerk_Palm(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,21}, 'omitnan');
            mean_Ang_Jerk_Palm(i+1,file)=mean(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,22}, 'omitnan');

            % STANDARD DEVIATION - Manual Handling
            std_PIP_Thumb(i+1,file)=std(Data_Acquisition{file}{(i*windowWidth)+1
: (i*windowWidth)+windowWidth,1}, 'omitnan');
            std_PIP_Index(i+1,file)=std(Data_Acquisition{file}{(i*windowWidth)+1
: (i*windowWidth)+windowWidth,2}, 'omitnan');
            std_PIP_Middle(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,3}, 'omitnan');
            std_MCP_FE_Thumb(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,6}, 'omitnan');
            std_MCP_FE_Index(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,7}, 'omitnan');
            std_MCP_FE_Middle(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,8}, 'omitnan');

```

Appendix B

```

        std_MCP_AA_Thumb(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,11}, 'omitnan');
        std_MCP_AA_Index(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,12}, 'omitnan');
        std_MCP_AA_Middle(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,13}, 'omitnan');
        std_MCP_IER_Thumb(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,16}, 'omitnan');
        std_FE_Wrist(i+1,file)=std(Data_Acquisition{file}{(i*windowWidth)+1
: (i*windowWidth)+windowWidth,17}, 'omitnan');
        std_AA_Wrist(i+1,file)=std(Data_Acquisition{file}{(i*windowWidth)+1
: (i*windowWidth)+windowWidth,18}, 'omitnan');
        std_Ang_Acc_Palm(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,19}, 'omitnan');
        std_Lin_Acc_Palm(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,20}, 'omitnan');
        std_Lin_Jerk_Palm(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,21}, 'omitnan');
        std_Ang_Jerk_Palm(i+1,file)=std(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,22}, 'omitnan');

% MEDIAN
        median_PIP_Thumb(i+1,file)=median(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,1}, 'omitnan');
        median_PIP_Index(i+1,file)=median(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,2}, 'omitnan');
        median_PIP_Middle(i+1,file)=median(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,3}, 'omitnan');
        median_MCP_FE_Thumb(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,6}, 'omitnan');
        median_MCP_FE_Index(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,7}, 'omitnan');
        median_MCP_FE_Middle(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,8}, 'omitnan');
        median_MCP_AA_Thumb(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,11}, 'omitnan');
        median_MCP_AA_Index(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,12}, 'omitnan');
        median_MCP_AA_Middle(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,13}, 'omitnan');
        median_MCP_IER_Thumb(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,16}, 'omitnan');
        median_FE_Wrist(i+1,file)=median(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,17}, 'omitnan');
        median_AA_Wrist(i+1,file)=median(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,18}, 'omitnan');
        median_Ang_Acc_Palm(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,19}, 'omitnan');
        median_Lin_Acc_Palm(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,20}, 'omitnan');
        median_Lin_Jerk_Palm(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,21}, 'omitnan');
        median_Ang_Jerk_Palm(i+1,file)=median(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,22}, 'omitnan');

% INTERQUARTILE RANGE
        iqr_PIP_Thumb(i+1,file)=iqr(Data_Acquisition{file}{(i*windowWidth)+1
: (i*windowWidth)+windowWidth,1});
        iqr_PIP_Index(i+1,file)=iqr(Data_Acquisition{file}{(i*windowWidth)+1
: (i*windowWidth)+windowWidth,2});
        iqr_PIP_Middle(i+1,file)=iqr(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,3});

```

Appendix B

```

        iqr_MCP_FE_Thumb(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,6});
        iqr_MCP_FE_Index(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,7});
        iqr_MCP_FE_Middle(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,8});
        iqr_MCP_AA_Thumb(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,11});
        iqr_MCP_AA_Index(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,12});
        iqr_MCP_AA_Middle(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,13});
        iqr_MCP_IER_Thumb(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,16});
        iqr_FE_Wrist(i+1,file)=iqr(Data_Acquisition{file}){(*windowWidth)+1
: (*windowWidth)+windowWidth,17});
        iqr_AA_Wrist(i+1,file)=iqr(Data_Acquisition{file}){(*windowWidth)+1
: (*windowWidth)+windowWidth,18});
        iqr_Ang_Acc_Palm(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,19});
        iqr_Lin_Acc_Palm(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,20});
        iqr_Lin_Jerk_Palm(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,21});
        iqr_Ang_Jerk_Palm(i+1,file)=iqr(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,22});

% COVARIANCE
        cov_PIP_Thumb(i+1,file)=cov(Data_Acquisition{file}){(*windowWidth)+1
: (*windowWidth)+windowWidth,1}, 'omitrows');
        cov_PIP_Index(i+1,file)=cov(Data_Acquisition{file}){(*windowWidth)+1
: (*windowWidth)+windowWidth,2}, 'omitrows');
        cov_PIP_Middle(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,3}, 'omitrows');
        cov_MCP_FE_Thumb(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,6}, 'omitrows');
        cov_MCP_FE_Index(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,7}, 'omitrows');
        cov_MCP_FE_Middle(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,8}, 'omitrows');
        cov_MCP_AA_Thumb(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,11}, 'omitrows');
        cov_MCP_AA_Index(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,12}, 'omitrows');
        cov_MCP_AA_Middle(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,13}, 'omitrows');
        cov_MCP_IER_Thumb(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,16}, 'omitrows');
        cov_FE_Wrist(i+1,file)=cov(Data_Acquisition{file}){(*windowWidth)+1
: (*windowWidth)+windowWidth,17}, 'omitrows');
        cov_AA_Wrist(i+1,file)=cov(Data_Acquisition{file}){(*windowWidth)+1
: (*windowWidth)+windowWidth,18}, 'omitrows');
        cov_Ang_Acc_Palm(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,19}, 'omitrows');
        cov_Lin_Acc_Palm(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,20}, 'omitrows');
        cov_Lin_Jerk_Palm(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,21}, 'omitrows');
        cov_Ang_Jerk_Palm(i+1,file)=cov(Data_Acquisition{file}){(*window-
Width)+1 : (*windowWidth)+windowWidth,22}, 'omitrows');

% ENTROPY

```

Appendix B

```

entropy_PIP_Thumb(i+1,file)=entropy(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,1});
entropy_PIP_Index(i+1,file)=entropy(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,2});
entropy_PIP_Middle(i+1,file)=entropy(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,3});
entropy_MCP_FE_Thumb(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,6});
entropy_MCP_FE_Index(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,7});
entropy_MCP_FE_Middle(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,8});
entropy_MCP_AA_Thumb(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,11});
entropy_MCP_AA_Index(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,12});
entropy_MCP_AA_Middle(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,13});
entropy_MCP_IER_Thumb(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,16});
entropy_FE_Wrist(i+1,file)=entropy(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,17});
entropy_AA_Wrist(i+1,file)=entropy(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,18});
entropy_Ang_Acc_Palm(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,19});
entropy_Lin_Acc_Palm(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,20});
entropy_Lin_Jerk_Palm(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,21});
entropy_Ang_Jerk_Palm(i+1,file)=entropy(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,22});

% SHAPE FACTOR
SF_PIP_Thumb(i+1,file) =
sqrt(mean_PIP_Thumb(i+1,file))/mean_PIP_Thumb(i+1,file);
SF_PIP_Index(i+1,file) = sqrt(mean_PIP_Index(i+1,file))/mean_PIP_In-
dex(i+1,file);
SF_PIP_Middle(i+1,file) = sqrt(mean_PIP_Mid-
dle(i+1,file))/mean_PIP_Middle(i+1,file);
SF_MCP_FE_Thumb(i+1,file) =
sqrt(mean_MCP_FE_Thumb(i+1,file))/mean_MCP_FE_Thumb(i+1,file);
SF_MCP_FE_Index(i+1,file) = sqrt(mean_MCP_FE_In-
dex(i+1,file))/mean_MCP_FE_Index(i+1,file);
SF_MCP_FE_Middle(i+1,file) = sqrt(mean_MCP_FE_Mid-
dle(i+1,file))/mean_MCP_FE_Middle(i+1,file);
SF_MCP_AA_Thumb(i+1,file) =
sqrt(mean_MCP_AA_Thumb(i+1,file))/mean_MCP_AA_Thumb(i+1,file);
SF_MCP_AA_Index(i+1,file) = sqrt(mean_MCP_AA_In-
dex(i+1,file))/mean_MCP_AA_Index(i+1,file);
SF_MCP_AA_Middle(i+1,file) = sqrt(mean_MCP_AA_Mid-
dle(i+1,file))/mean_MCP_AA_Middle(i+1,file);
SF_MCP_IER_Thumb(i+1,file) =
sqrt(mean_MCP_IER_Thumb(i+1,file))/mean_MCP_IER_Thumb(i+1,file);
SF_FE_Wrist(i+1,file) =
sqrt(mean_FE_Wrist(i+1,file))/mean_FE_Wrist(i+1,file);
SF_AA_Wrist(i+1,file) =
sqrt(mean_AA_Wrist(i+1,file))/mean_AA_Wrist(i+1,file);
SF_Lin_Acc_Palm(i+1,file) =
sqrt(mean_Lin_Acc_Palm(i+1,file))/mean_Lin_Acc_Palm(i+1,file);
SF_Ang_Acc_Palm(i+1,file) =
sqrt(mean_Ang_Acc_Palm(i+1,file))/mean_Ang_Acc_Palm(i+1,file);

```

Appendix B

```

        SF_Lin_Jerk_Palm(i+1,file) =
sqrt(mean_Lin_Jerk_Palm(i+1,file))/mean_Lin_Jerk_Palm(i+1,file);
        SF_Ang_Jerk_Palm(i+1,file) =
sqrt(mean_Ang_Jerk_Palm(i+1,file))/mean_Ang_Jerk_Palm(i+1,file);

        % PEAK to PEAK
        P2P_PIP_Thumb(i+1,file) = peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,1});
        P2P_PIP_Index(i+1,file) = peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,2});
        P2P_PIP_Middle(i+1,file) = peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,3});
        P2P_MCP_FE_Thumb(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,6});
        P2P_MCP_FE_Index(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,7});
        P2P_MCP_FE_Middle(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,8});
        P2P_MCP_AA_Thumb(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,11});
        P2P_MCP_AA_Index(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,12});
        P2P_MCP_AA_Middle(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,13});
        P2P_MCP_IER_Thumb(i+1,file) = peak2peak(Data_Acquisi-
tion{file}{(i*windowWidth)+1 : (i*windowWidth)+windowWidth,16});
        P2P_FE_Wrist(i+1,file) = peak2peak(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,17});
        P2P_AA_Wrist(i+1,file) = peak2peak(Data_Acquisition{file}{(i*window-
Width)+1 : (i*windowWidth)+windowWidth,18});
        P2P_Ang_Acc_Palm(i+1,file)=peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,19});
        P2P_Lin_Acc_Palm(i+1,file)=peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,20});
        P2P_Lin_Jerk_Palm(i+1,file)=peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,21});
        P2P_Ang_Jerk_Palm(i+1,file)=peak2peak(Data_Acquisition{file}{(i*win-
dowWidth)+1 : (i*windowWidth)+windowWidth,22});
    end

    %% Peak-related Features
    % Find Peaks that exceed 60% of the max peak and are at least 0.5
    % second apart
    frac = 0.6; % 60%
    MinDistance = 0.5; % 0.5 second

    [y_PIP_Thumb, x_PIP_Thumb] = findpeaks(Data_Acquisi-
tion{file}{:,1},time,'MinPeakHeight', ...
        max(Data_Acquisition{file}{:,1})*frac,'MinPeakDistance', MinDistance);
    [y_PIP_Index, x_PIP_Index] = findpeaks(Data_Acquisi-
tion{file}{:,2},time,'MinPeakHeight', ...
        max(Data_Acquisition{file}{:,2})*frac,'MinPeakDistance', MinDistance);
    [y_PIP_Middle, x_PIP_Middle] = findpeaks(Data_Acquisi-
tion{file}{:,3},time,'MinPeakHeight', ...
        max(Data_Acquisition{file}{:,3})*frac,'MinPeakDistance', MinDistance);
    [y_MCP_FE_Thumb, x_MCP_FE_Thumb] = findpeaks(Data_Acquisi-
tion{file}{:,6},time,'MinPeakHeight', ...
        max(Data_Acquisition{file}{:,6})*frac,'MinPeakDistance', MinDistance);
    [y_MCP_FE_Index, x_MCP_FE_Index] = findpeaks(Data_Acquisi-
tion{file}{:,7},time,'MinPeakHeight', ...
        max(Data_Acquisition{file}{:,7})*frac,'MinPeakDistance', MinDistance);

```

Appendix B

```

[y_MCP_FE_Middle, x_MCP_FE_Middle] = findpeaks(Data_Acquisi-
tion{file}{:,8},time,'MinPeakHeight', ...
max(Data_Acquisition{file}{:,8})*frac,'MinPeakDistance', MinDistance);
[y_FE_Wrist, x_FE_Wrist] = findpeaks(Data_Acquisi-
tion{file}{:,17},time,'MinPeakHeight', ...
max(Data_Acquisition{file}{:,17})*frac,'MinPeakDistance', MinDistance);
[y_Lin_Acc_Palm, x_Lin_Acc_Palm] = findpeaks(Data_Acquisi-
tion{file}{:,20},time,'MinPeakHeight', ...
max(Data_Acquisition{file}{:,20})*frac,'MinPeakDistance', MinDistance);
[y_Ang_Acc_Palm, x_Ang_Acc_Palm] = findpeaks(Data_Acquisi-
tion{file}{:,19},time,'MinPeakHeight', ...
max(Data_Acquisition{file}{:,19})*frac,'MinPeakDistance', MinDistance);

%% Count number of peaks every windowWidth
for i=1:floor(size(Data_Acquisition{file},1)/windowWidth)
    par_peaks_PIP_Thumb(i,:) = x_PIP_Thumb <= (i*(windowWidth/SampTime));%
    this produces a binary array
    par_peaks_PIP_Index(i,:) = x_PIP_Index <= (i*(windowWidth/SampTime));
    par_peaks_PIP_Middle(i,:) = x_PIP_Middle <= (i*(windowWidth/SampTime));
    % par_peaks_PIP_Ring(i,:) = x_PIP_Ring <= (i*(windowWidth/SampTime));
    % par_peaks_PIP_Little(i,:) = x_PIP_Little <= (i*(window-
    Width/SampTime));
    par_peaks_MCP_FE_Thumb(i,:) = x_MCP_FE_Thumb <= (i*(window-
    Width/SampTime));
    par_peaks_MCP_FE_Index(i,:) = x_MCP_FE_Index <= (i*(window-
    Width/SampTime));
    par_peaks_MCP_FE_Middle(i,:) = x_MCP_FE_Middle <= (i*(window-
    Width/SampTime));
    % par_peaks_MCP_FE_Ring(i,:) = x_MCP_FE_Ring <= (i*(window-
    Width/SampTime));
    % par_peaks_MCP_FE_Little(i,:) = x_MCP_FE_Little <= (i*(window-
    Width/SampTime));
    par_peaks_FE_Wrist(i,:) = x_FE_Wrist <= (i*(windowWidth/SampTime));
    par_peaks_Lin_Acc_Palm(i,:) = x_Lin_Acc_Palm <= (i*(window-
    Width/SampTime));
    par_peaks_Ang_Acc_Palm(i,:) = x_Ang_Acc_Palm <= (i*(window-
    Width/SampTime));
    if i==1
        peaks_PIP_Thumb(i,file)=sum(par_peaks_PIP_Thumb(i,:));
        peaks_PIP_Index(i,file)=sum(par_peaks_PIP_Index(i,:));
        peaks_PIP_Middle(i,file)=sum(par_peaks_PIP_Middle(i,:));
        % peaks_PIP_Ring(i,file)=sum(par_peaks_PIP_Ring(i,:));
        % peaks_PIP_Little(i,file)=sum(par_peaks_PIP_Little(i,:));
        peaks_MCP_FE_Thumb(i,file)=sum(par_peaks_MCP_FE_Thumb(i,:));
        peaks_MCP_FE_Index(i,file)=sum(par_peaks_MCP_FE_Index(i,:));
        peaks_MCP_FE_Middle(i,file)=sum(par_peaks_MCP_FE_Middle(i,:));
        % peaks_MCP_FE_Ring(i,file)=sum(par_peaks_MCP_FE_Ring(i,:));
        % peaks_MCP_FE_Little(i,file)=sum(par_peaks_MCP_FE_Little(i,:));
        peaks_FE_Wrist(i,file)=sum(par_peaks_FE_Wrist(i,:));
        peaks_Lin_Acc_Palm(i,file)=sum(par_peaks_Lin_Acc_Palm(i,:));
        peaks_Ang_Acc_Palm(i,file)=sum(par_peaks_Ang_Acc_Palm(i,:));
    else
        peaks_PIP_Thumb(i,file)=sum(par_peaks_PIP_Thumb(i,:)) -
        sum(peaks_PIP_Thumb(1:i-1,file));
        peaks_PIP_Index(i,file)=sum(par_peaks_PIP_Index(i,:)) -
        sum(peaks_PIP_Index(1:i-1,file));
        peaks_PIP_Middle(i,file)=sum(par_peaks_PIP_Middle(i,:)) -
        sum(peaks_PIP_Middle(1:i-1,file));
        % peaks_PIP_Ring(i,file)=sum(par_peaks_PIP_Ring(i,:)) -
        sum(peaks_PIP_Ring(1:i-1,file));
        % peaks_PIP_Little(i,file)=sum(par_peaks_PIP_Little(i,:)) -
        sum(peaks_PIP_Little(1:i-1,file));
    end
end

```

Appendix B

```

        peaks_MCP_FE_Thumb(i,file)=sum(par_peaks_MCP_FE_Thumb(i,:)) -
sum(peaks_MCP_FE_Thumb(1:i-1,file));
        peaks_MCP_FE_Index(i,file)=sum(par_peaks_MCP_FE_Index(i,:)) -
sum(peaks_MCP_FE_Index(1:i-1,file));
        peaks_MCP_FE_Middle(i,file)=sum(par_peaks_MCP_FE_Middle(i,:)) -
sum(peaks_MCP_FE_Middle(1:i-1,file));
%       peaks_MCP_FE_Ring(i,file)=sum(par_peaks_MCP_FE_Ring(i,:)) -
sum(peaks_MCP_FE_Ring(1:i-1,file));
%       peaks_MCP_FE_Little(i,file)=sum(par_peaks_MCP_FE_Little(i,:)) -
sum(peaks_MCP_FE_Little(1:i-1,file));
        peaks_FE_Wrist(i,file)=sum(par_peaks_FE_Wrist(i,:)) -
sum(peaks_FE_Wrist(1:i-1,file));
        peaks_Lin_Acc_Palm(i,file)=sum(par_peaks_Lin_Acc_Palm(i,:)) -
sum(peaks_Lin_Acc_Palm(1:i-1,file));
        peaks_Ang_Acc_Palm(i,file)=sum(par_peaks_Ang_Acc_Palm(i,:)) -
sum(peaks_Ang_Acc_Palm(1:i-1,file));
    end
end
clear par_peaks_PIP_Thumb par_peaks_PIP_Index par_peaks_PIP_Middle
par_peaks_PIP_Ring par_peaks_PIP_Little
clear par_peaks_MCP_FE_Thumb par_peaks_MCP_FE_Index par_peaks_MCP_FE_Middle
par_peaks_MCP_FE_Ring par_peaks_MCP_FE_Little
clear par_peaks_FE_Wrist par_peaks_Lin_Acc_Palm par_peaks_Ang_Acc_Palm

%% Results
Task_name = {Task_list(file).name(1:end-9)} + strings(i,1);
Features = table(mean_PIP_Thumb(1:i,file), mean_PIP_Index(1:i,file),
mean_PIP_Middle(1:i,file), ...
    mean_MCP_FE_Thumb(1:i,file), mean_MCP_FE_Index(1:i,file),
mean_MCP_FE_Middle(1:i,file), ...
    mean_MCP_AA_Thumb(1:i,file), mean_MCP_AA_Index(1:i,file),
mean_MCP_AA_Middle(1:i,file), ...
    mean_MCP_IER_Thumb(1:i,file), mean_FE_Wrist(1:i,file),
mean_AA_Wrist(1:i,file), ...
    mean_Lin_Acc_Palm(1:i,file), mean_Ang_Acc_Palm(1:i,file),
mean_Lin_Jerk_Palm(1:i,file), mean_Ang_Jerk_Palm(1:i,file), ...
    std_PIP_Thumb(1:i,file), std_PIP_Index(1:i,file), std_PIP_Mid-
dle(1:i,file), ...
    std_MCP_FE_Thumb(1:i,file), std_MCP_FE_Index(1:i,file),
std_MCP_FE_Middle(1:i,file), ...
    std_MCP_AA_Thumb(1:i,file), std_MCP_AA_Index(1:i,file),
std_MCP_AA_Middle(1:i,file), ...
    std_MCP_IER_Thumb(1:i,file), std_FE_Wrist(1:i,file),
std_AA_Wrist(1:i,file), std_Lin_Acc_Palm(1:i,file), std_Ang_Acc_Palm(1:i,file),
std_Lin_Jerk_Palm(1:i,file), std_Ang_Jerk_Palm(1:i,file),...
    median_PIP_Thumb(1:i,file), median_PIP_Index(1:i,file), me-
dian_PIP_Middle(1:i,file), ...
    median_MCP_FE_Thumb(1:i,file), median_MCP_FE_Index(1:i,file), me-
dian_MCP_FE_Middle(1:i,file), ...
    median_MCP_AA_Thumb(1:i,file), median_MCP_AA_Index(1:i,file), me-
dian_MCP_AA_Middle(1:i,file), ...
    median_MCP_IER_Thumb(1:i,file), median_FE_Wrist(1:i,file), me-
dian_AA_Wrist(1:i,file), median_Lin_Acc_Palm(1:i,file), me-
dian_Ang_Acc_Palm(1:i,file), median_Lin_Jerk_Palm(1:i,file), me-
dian_Ang_Jerk_Palm(1:i,file),...
    iqr_PIP_Thumb(1:i,file), iqr_PIP_Index(1:i,file), iqr_PIP_Mid-
dle(1:i,file), ...
    iqr_MCP_FE_Thumb(1:i,file), iqr_MCP_FE_Index(1:i,file),
iqr_MCP_FE_Middle(1:i,file), ...
    iqr_MCP_AA_Thumb(1:i,file), iqr_MCP_AA_Index(1:i,file),
iqr_MCP_AA_Middle(1:i,file), ...

```

Appendix B

```

iqr_MCP_IER_Thumb(1:i,file), iqr_FE_Wrist(1:i,file),
iqr_AA_Wrist(1:i,file), iqr_Lin_Acc_Palm(1:i,file), iqr_Ang_Acc_Palm(1:i,file),
iqr_Lin_Jerk_Palm(1:i,file), iqr_Ang_Jerk_Palm(1:i,file),...
cov_PIP_Thumb(1:i,file), cov_PIP_Index(1:i,file), cov_PIP_Mid-
dle(1:i,file), ...
cov_MCP_FE_Thumb(1:i,file), cov_MCP_FE_Index(1:i,file),
cov_MCP_FE_Middle(1:i,file), ...
cov_MCP_AA_Thumb(1:i,file), cov_MCP_AA_Index(1:i,file),
cov_MCP_AA_Middle(1:i,file), ...
cov_MCP_IER_Thumb(1:i,file), cov_FE_Wrist(1:i,file),
cov_AA_Wrist(1:i,file), cov_Lin_Acc_Palm(1:i,file), cov_Ang_Acc_Palm(1:i,file),
cov_Lin_Jerk_Palm(1:i,file), cov_Ang_Jerk_Palm(1:i,file),...
entropy_PIP_Thumb(1:i,file), entropy_PIP_Index(1:i,file), en-
tropy_PIP_Middle(1:i,file), ...
entropy_MCP_FE_Thumb(1:i,file), entropy_MCP_FE_Index(1:i,file), en-
tropy_MCP_FE_Middle(1:i,file), ...
entropy_MCP_AA_Thumb(1:i,file), entropy_MCP_AA_Index(1:i,file), en-
tropy_MCP_AA_Middle(1:i,file), ...
entropy_MCP_IER_Thumb(1:i,file), entropy_FE_Wrist(1:i,file), en-
tropy_AA_Wrist(1:i,file), entropy_Lin_Acc_Palm(1:i,file), en-
tropy_Ang_Acc_Palm(1:i,file), entropy_Lin_Jerk_Palm(1:i,file), en-
tropy_Ang_Jerk_Palm(1:i,file),...
SF_PIP_Thumb(1:i,file), SF_PIP_Index(1:i,file), SF_PIP_Mid-
dle(1:i,file), ...
SF_MCP_FE_Thumb(1:i,file), SF_MCP_FE_Index(1:i,file), SF_MCP_FE_Mid-
dle(1:i,file), ...
SF_MCP_AA_Thumb(1:i,file), SF_MCP_AA_Index(1:i,file), SF_MCP_AA_Mid-
dle(1:i,file), ...
SF_MCP_IER_Thumb(1:i,file), SF_FE_Wrist(1:i,file),
SF_AA_Wrist(1:i,file), SF_Lin_Acc_Palm(1:i,file), SF_Ang_Acc_Palm(1:i,file),
SF_Lin_Jerk_Palm(1:i,file), SF_Ang_Jerk_Palm(1:i,file),...
P2P_PIP_Thumb(1:i,file), P2P_PIP_Index(1:i,file), P2P_PIP_Mid-
dle(1:i,file), ...
P2P_MCP_FE_Thumb(1:i,file), P2P_MCP_FE_Index(1:i,file),
P2P_MCP_FE_Middle(1:i,file), ...
P2P_MCP_AA_Thumb(1:i,file), P2P_MCP_AA_Index(1:i,file),
P2P_MCP_AA_Middle(1:i,file), ...
P2P_MCP_IER_Thumb(1:i,file), P2P_FE_Wrist(1:i,file),
P2P_AA_Wrist(1:i,file), P2P_Lin_Acc_Palm(1:i,file), P2P_Ang_Acc_Palm(1:i,file),
P2P_Lin_Jerk_Palm(1:i,file), P2P_Ang_Jerk_Palm(1:i,file),...
peaks_PIP_Thumb(1:i,file), peaks_PIP_Index(1:i,file), peaks_PIP_Mid-
dle(1:i,file), ...
peaks_MCP_FE_Thumb(1:i,file), peaks_MCP_FE_Index(1:i,file),
peaks_MCP_FE_Middle(1:i,file), ...
peaks_FE_Wrist(1:i,file), peaks_Lin_Acc_Palm(1:i,file),
peaks_Ang_Acc_Palm(1:i,file), ...
Task_name, 'VariableNames', varNames);

% Write tables
writetable(Features, strcat('Database_Features\GLOVE\Sub', num2str(sub-
jects), '\', Task_list(file).name));
end
clear mean_PIP_Thumb mean_PIP_Index mean_PIP_Middle mean_MCP_FE_Thumb
mean_MCP_FE_Index mean_MCP_FE_Middle ...
mean_MCP_AA_Thumb mean_MCP_AA_Index mean_MCP_AA_Middle
mean_MCP_IER_Thumb mean_FE_Wrist mean_AA_Wrist ...
mean_Lin_Acc_Palm mean_Ang_Acc_Palm mean_Lin_Jerk_Palm
mean_Ang_Jerk_Palm ...
std_PIP_Thumb std_PIP_Index std_PIP_Middle std_MCP_FE_Thumb
std_MCP_FE_Index std_MCP_FE_Middle ...
std_MCP_AA_Thumb std_MCP_AA_Index std_MCP_AA_Middle
std_MCP_IER_Thumb std_FE_Wrist std_AA_Wrist ...

```


Appendix B

```

std_Lin_Acc_Palm std_Ang_Acc_Palm std_Lin_Jerk_Palm
std_Ang_Jerk_Palm...
median_PIP_Thumb median_PIP_Index median_PIP_Middle ...
median_MCP_FE_Thumb median_MCP_FE_Index median_MCP_FE_Middle ...
median_MCP_AA_Thumb median_MCP_AA_Index median_MCP_AA_Middle ...
median_MCP_IER_Thumb median_FE_Wrist median_AA_Wrist me-
dian_Lin_Acc_Palm median_Ang_Acc_Palm median_Lin_Jerk_Palm me-
dian_Ang_Jerk_Palm...
iqr_PIP_Thumb iqr_PIP_Index iqr_PIP_Middle ...
iqr_MCP_FE_Thumb iqr_MCP_FE_Index iqr_MCP_FE_Middle ...
iqr_MCP_AA_Thumb iqr_MCP_AA_Index iqr_MCP_AA_Middle ...
iqr_MCP_IER_Thumb iqr_FE_Wrist iqr_AA_Wrist iqr_Lin_Acc_Palm
iqr_Ang_Acc_Palm iqr_Lin_Jerk_Palm iqr_Ang_Jerk_Palm...
cov_PIP_Thumb cov_PIP_Index cov_PIP_Middle ...
cov_MCP_FE_Thumb cov_MCP_FE_Index cov_MCP_FE_Middle ...
cov_MCP_AA_Thumb cov_MCP_AA_Index cov_MCP_AA_Middle ...
cov_MCP_IER_Thumb cov_FE_Wrist cov_AA_Wrist cov_Lin_Acc_Palm
cov_Ang_Acc_Palm cov_Lin_Jerk_Palm cov_Ang_Jerk_Palm...
entropy_PIP_Thumb entropy_PIP_Index entropy_PIP_Middle ...
entropy_MCP_FE_Thumb entropy_MCP_FE_Index entropy_MCP_FE_Middle ...
entropy_MCP_AA_Thumb entropy_MCP_AA_Index entropy_MCP_AA_Middle ...
entropy_MCP_IER_Thumb entropy_FE_Wrist entropy_AA_Wrist en-
tropy_Lin_Acc_Palm entropy_Ang_Acc_Palm entropy_Lin_Jerk_Palm en-
tropy_Ang_Jerk_Palm...
SF_PIP_Thumb SF_PIP_Index SF_PIP_Middle ...
SF_MCP_FE_Thumb SF_MCP_FE_Index SF_MCP_FE_Middle ...
SF_MCP_AA_Thumb SF_MCP_AA_Index SF_MCP_AA_Middle ...
SF_MCP_IER_Thumb SF_FE_Wrist SF_AA_Wrist SF_Lin_Acc_Palm
SF_Ang_Acc_Palm SF_Lin_Jerk_Palm SF_Ang_Jerk_Palm...
P2P_PIP_Thumb P2P_PIP_Index P2P_PIP_Middle ...
P2P_MCP_FE_Thumb P2P_MCP_FE_Index P2P_MCP_FE_Middle ...
P2P_MCP_AA_Thumb P2P_MCP_AA_Index P2P_MCP_AA_Middle ...
P2P_MCP_IER_Thumb P2P_FE_Wrist P2P_AA_Wrist P2P_Lin_Acc_Palm
P2P_Ang_Acc_Palm P2P_Lin_Jerk_Palm P2P_Ang_Jerk_Palm...
peaks_PIP_Thumb peaks_PIP_Index peaks_PIP_Middle ...
peaks_MCP_FE_Thumb peaks_MCP_FE_Index peaks_MCP_FE_Middle ...
peaks_FE_Wrist peaks_Lin_Acc_Palm peaks_Ang_Acc_Palm
end

```