# POLITECNICO DI TORINO

**Master of Science's Degree in Biomedical Engineering**

**Master of Science's Degree Thesis**

# Capturing Gait Signature: A Preliminary Study on the Feasibility of Recognizing People by their Gait Based on a Biomechanically-driven Marker-less Approach Using Multiple RGB Cameras.

**Supervisors**
**Prof. Andrea Cereatti**
**Ing. Diletta Balta**
**Ing. Paolo Tasca**

**Candidate**
**Benedetta SABBADINI**

**October 2024**

# Abstract

Gait recognition based on video data has gained popularity in both surveillance and clinical contexts due to its unique ability to identify individuals based on walking patterns. Unlike other biometrics like fingerprints or facial recognition, gait works effectively from a distance, does not need user cooperation, handles low-quality videos, remains reliable when body traits are hidden, and is hard to imitate. Gait recognition uses either model-free or model-based methods. Model-free approaches rely on image-based features, while model-based methods focus on biomechanical features like joint kinematics. Model-based methods are easier to understand, view- and scale-invariant, and less affected by background noise, but they have greater computational complexity. Identifying the most relevant gait features is essential to reduce this load. This work proposes a model-based, marker-less method for gait recognition driven by biomechanical features. To reduce computational complexity, features were selected automatically from a wide range of parameters from the literature, covering various domains and modalities. The study involved fifteen healthy subjects walking at three different speeds along a 5m indoor walkway, recorded by three RGB cameras. Joint centres were tracked semi-automatically using MoveNet, a deep-learning pose estimator. Stride segmentation was automatically performed to extract gait features stride-by-stride, including both time-domain and frequency-domain features. Features were ranked using the minimum redundancy maximum relevance algorithm, and those scoring below 0.001 were excluded, resulting in 19 features, including kinematic, frequency parameters, and correlation indices. Then, top-down wrapper feature selection was applied using seven classification models: (i) Decision Tree, (ii) Discriminant

Analysis, (iii) Ensemble Classifier, (iv) ECOC, (v) k-NN, (vi) Naïve Bayes, and (vii) Neural Networks. Classification accuracy was assessed via 5-fold cross-validation, and the best feature set was chosen for each model. Hyperparameters were tuned by grid-search and 5-fold cross-validation, and the best-performing configuration was selected for each model. Finally, models were evaluated on unseen data, with recall, precision, and F1-score metrics calculated for the entire dataset and each walking speed. The ECOC model achieved the highest F1-score (78.71%), followed by k-NN (73.40%), while Decision Tree performed poorly (39.86%) due to overfitting, sensitivity to small changes, and reliance on single-feature splits, limiting its ability to capture complex gait patterns. Other models' F1-scores were 69.58% (Discriminant Analysis), 64.15% (Ensemble Classifier), 62.08% (Naïve Bayes), and 66.19% (Neural Networks). Although this method performed lower than previous studies, likely due to dataset homogeneity, the identified feature subsets provide a strong basis for discriminating individuals from their gait. Performance could improve with a larger, more diverse dataset and varied gait patterns. This study contributes to the development of biomechanically driven gait recognition with potential applications in clinical assessment and video surveillance.

# Acknowledgements

At the end of this journey, I would like to extend my sincere gratitude to Professor Cereatti for providing me with the chance to undertake this project and for allowing me to play an active role in its execution.

Moreover, I have to express my immense gratitude to Ing. Diletta Balta and Ing. Paolo Tasca for the help and the support they gave to me from the beginning to the end of this thesis project. Your advice and availability have been fundamental to me during this year. Thank you from the bottom of my heart.

I want to acknowledge the colleagues I’ve met during this experience. It wouldn’t have been the same without you.

Last but not least, I would like to thank all my family, dad, mom, Leti and Tommi, my grandparents and all my friends for their constant support and for never letting me give up when everything seemed impossible. Your unconditional love has been the engine of my journey.

# Contents

# Acronyms

**AC** acromioclavicular joint

**ASA** Absolute Asymmetry Angle

**CCS** Cartesian Coordinate System

**CM** Confusion Matrix

**CS** Construction Set

**ECOC** Error Correcting Output Codes

**FWHM** Full-Width at Half Maximum

**GA** Gait Asymmetry

**GC** Gait cycle

**GH** glenohumeral joint

**IC** Initial Contact

**IQR** Interquartile range

**IR** Infrared Ligth

**JCS** Joint Coordinates System

**k-NN** k-Nearest Neighbors

**LED** Light-Emitting Diode

**ML** Machine Learning

**MOCAP** Motion Capture

**mRMR** Minimum Redundancy Maximum Relevance

**PSD** Power Spectral Density

**RBF** Radial Basis Function

**ST** scapulothoracic joint

**TRS** Training Set

**TS** Test Set

**VS** Validation Set

# 1 Introduction

## 1.1 Motivation and general introduction

Biometrics is defined as “the science of establishing the identity of an individual, based on his/her inherent physical and behavioural traits” [1]. Nowadays biometric systems become increasingly vital for security and identification, and traditional methods like fingerprint and facial recognition have significant limitations. Fingerprints can be altered or obscured due to wear or injuries, and facial recognition systems struggle with variations in lighting, angle, and expression. Moreover, both methods are vulnerable to attacks [2], such as the use of artificial fingerprints or 3D-printed masks, and the rise of Deepfake technology poses further risks by creating convincing fake identities. These vulnerabilities highlight the need to explore alternative biometric modalities that offer greater security and resilience.

Gait recognition, which identifies individuals based on their unique walking patterns, presents a promising solution. Gait has several distinct properties that make it a robust biometric trait: it is difficult to disguise or replicate, can be captured at a distance and with simple instrumentation, and does not require active participation from the subject. Additionally, gait remains consistent over time and is less affected by external factors like clothing or environment [3]. Moreover, gait recognition holds significant potential in healthcare [4]. The ability to monitor and analyse gait patterns remotely through video-based systems can be precious for early diagnosis of neurodegenerative diseases, such as Parkinson’s or Alzheimer’s, where changes in gait are often early indicators. Furthermore, video-based gait analysis can be used in rehabilitation to track patient progress and tailor treatment plans more effectively, providing a non-invasive method to monitor recovery in real-time.

With advancements in technology, automatic video-based gait recognition has emerged as a sophisticated method for leveraging this biometric. This thesis is motivated by the limitations of traditional biometric systems and seeks to investigate gait recognition as a potential alternative. By exploiting the distinct and stable characteristics of an individual's gait, this method could provide a more secure and reliable means of identification, while also opening new opportunities for healthcare applications.

## 1.2 Aim of the work and challenges

The aim of this study is to develop a model-based approach that leverages biomechanical features for the automatic recognition of gait patterns, using a multi-camera, marker-less system. A key challenge of this work is to design a method that is not only effective with marker-less systems but also adaptable or directly integrable to marker-based setups, and capable of being further developed for medical purposes. This adaptability ensures the method’s applicability across different technologies and environments, while paving the way for its use in healthcare, such as in the diagnosis and monitoring of gait-related disorders.

## 1.3 Thesis Outline

**Chapter 2** provides an in-depth analysis of gait cycle, covering its basic notions and taxonomy. Then, the chapter provides a general overview of gait analysis focusing on key gait parameters, such as spatio-temporal parameters and joint kinematics, it explores gait measurement technologies and the role of machine learning in gait studies. At the end, the chapter reviews the various systems used for capturing motion data, including marker-based and marker-less systems, and their respective advantages and limitations.

**Chapter 3** outlines the data collection process, and the experimental setup and protocol.

**Chapter 4** details the methods used in this thesis, beginning with the detection of joint centres and the gait cycle segmentation. It then covers feature extraction, including both time-domain and frequency-domain features, and the process of feature selection. The chapter also describes the classification models used in the study and the evaluation metrics employed to assess their performance.

**Chapter 5** displays the results obtained with the implemented methods focusing on classification performance.

**Chapter 6** discusses the results as well as the limitations of the study.

**Chapter 7** sums up the main achievements of the thesis and provides recommendations for future research.

# 2 Background

This chapter introduces the fundamental concepts of gait and gait analysis, establishing the terminology and foundational ideas that will be explored in the subsequent chapters.

## 2.1 Gait cycle

Gait cycle (GC) is defined as the sequence of events/movements between two consecutive contacts of the same foot with the ground. It is considered the fundamental unit of gait [5].

### 2.1.1 Basic notions

The crucial distinction between gait and walking must be noted since gait is referred to as the process of walking but is defined as the individualistic way of walking [6]. According to Perry [7], gait is not just a basic function but a complex, coordinated activity that reflects the overall health and function of the musculoskeletal and nervous systems. The alternating and repeated use of the lower limbs moves the body forward while also ensuring stable support. Human movement is facilitated through a highly coordinated interaction among the bones, muscles, ligaments, and joints within the musculoskeletal system, which is regulated by the nervous system. The central nervous system sends commands to the musculoskeletal system through the peripheral nervous system, enabling the generation of forces necessary for movement and the performance of daily activities.

The body can be divided into two functional units with respect to gait: the locomotive unit and the passenger unit. The locomotive unit, including the pelvis and lower limbs, uses 11 joints and 57 muscles to support and move the body forward. It serves four main functions: propulsion, stability, impact absorption, and energy conservation. The passenger unit, comprising the head, arms, and trunk (sometimes referred to as HAT[1]), accounts for 70% of body weight and maintains postural integrity. Its alignment influences lower limb muscle activity, while its mass is key to postural stability.

### 2.1.2 Taxonomy of gait

The foot-to-ground contact is the easiest to detect, so it is considered as the beginning of the gait cycle. This event is commonly known as *initial contact* (IC). The gait cycle is typically divided into two primary phases: stance and swing (Figure 2.1.2.1). The stance phase begins with the IC and accounts for approximately 60% of the gait cycle. Conversely, the swing phase, which constitutes about 40% of the gait cycle, starts with the toe off and refers to the period the foot is lifted off the ground and moves forward.

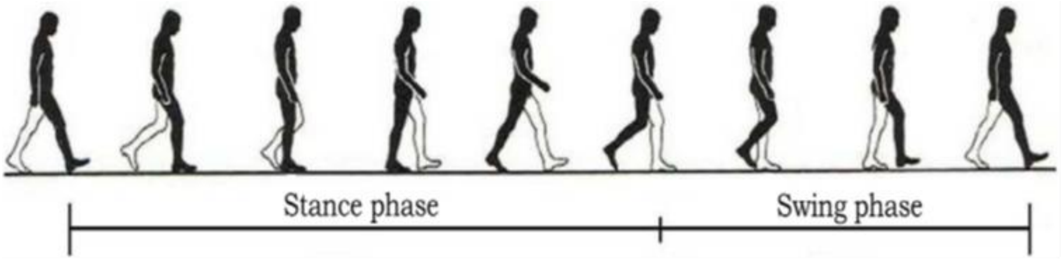


*Figure 2.1.2.1: The two main phases of gait according to Perry [7].*

At the start and end of the cycle, both feet contact the ground, so this period is known as *double support*. In contrast, during the middle of the stance phase, only one foot is in contact with the ground, which is referred to as *single support* (Figure 2.1.2.2).

---

[1] Head, Arms, Trunk.

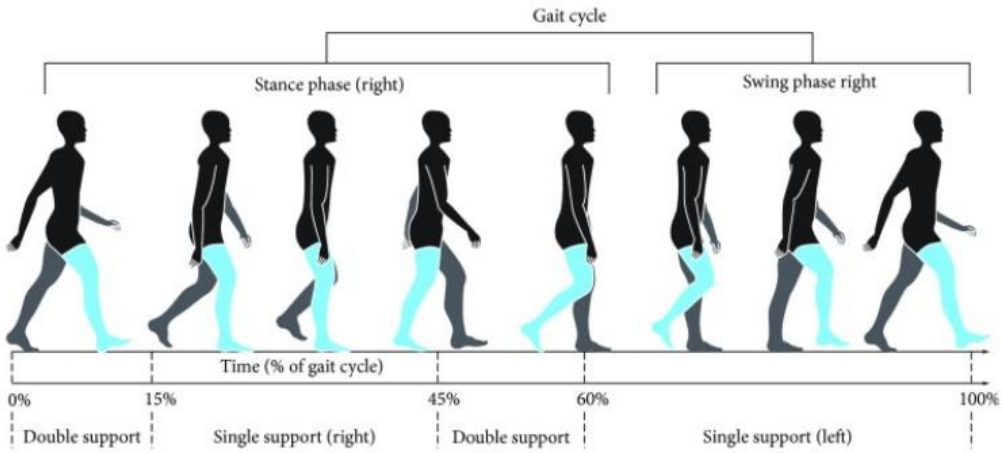


*Figure 2.1.2.2: Illustration of the single and double support stages.*

The gait cycle is also referred to as *stride* and can be considered as the sequence of two *steps* (Figure 2.1.2.3). A stride begins with the initial contact of one foot and ends with the next initial contact of the same foot. So, if the stride starts with a right step, it will be ended by a left step.

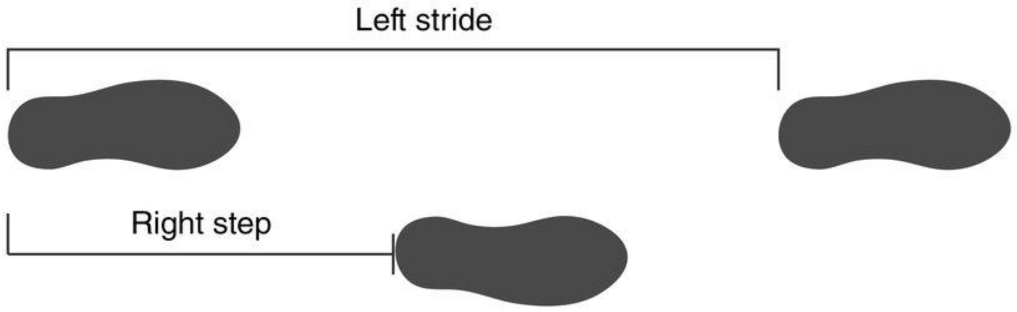


*Figure 2.1.2.3: Relationship between steps and stride.*

According to Perry [7], the gait cycle is further divided into a total of 8 sub-phases (Figure 2.1.2.4). The stance phase consists in five sub-phases:

1) Initial contact (heel strike): 0 – 2 % GC

   This phase marks the instant when the foot contacts the ground and starts the first part of double support, where both feet are touching the ground. Its primary purpose is to establish ground contact and initiate the acceptance of body weight. During this phase, the ground reaction force shifts behind the ankle joint, causing the foot to rotate in a clockwise direction.

2) Loading response (foot float): 2 – 10 % GC

   This phase starts with initial contact and continues until the contralateral foot leaves the ground. The foot continues to accept weight and absorb shock by rolling into pronation (first “rocker”).

3) Midstance: 10 – 30 % GC

   This phase begins when the opposite foot leaves the ground and continues until the heel of the current foot lifts off. The second rocker mechanism is observed during midstance, where the tibia rotates over the ankle joint and moves up and over the talus. The intrinsic muscles of the foot and the tibialis posterior work to support the medial longitudinal arch.

4) Terminal stance (heel off): 30 – 50 % GC

   This phase begins when the heel leaves the floor and continues until the contralateral foot contacts the ground. In addition to single limb support and stability, this event serves to propel the body forward. The bodyweight is distributed across the metatarsal heads. During terminal stance, the third and final rocker mechanism is activated as the foot undergoes plantar flexion over a stable forefoot, particularly at the metatarsophalangeal joints.

5) Pre-swing (toe off): 50 – 60 % GC

   This phase starts when the opposite foot contacts the ground and lasts until the current foot leaves the ground. It provides the final burst of propulsion as the toes lift off. During pre-swing, the weight shifts from one foot to the other, with both feet contacting the ground during this period, in contrast to the single support phase that predominates in other intervals.

The swing phase consists in three sub-phases:

1) Initial swing: 60 – 75 % GC

   This sub-phase begins when the foot lifts off the ground and continues until it aligns with the opposite ankle. Its primary purpose is to move the limb forward and shorten its length to ensure adequate foot clearance.

2) Mid-swing: 75 – 87 % GC

   Mid-swing starts when the ankle and foot of the swing leg align and extends until the tibia of the swing leg becomes vertical. Like the initial swing, this phase is focused on advancing the limb and shortening its length to facilitate foot clearance.

3) Terminal swing: 87 – 100 % GC

   Terminal swing begins when the swing leg's tibia is vertical and concludes with the initial contact of the foot. During this phase, the pelvis continues to flex due to the inertia of the leg, with minimal muscular contribution involved.

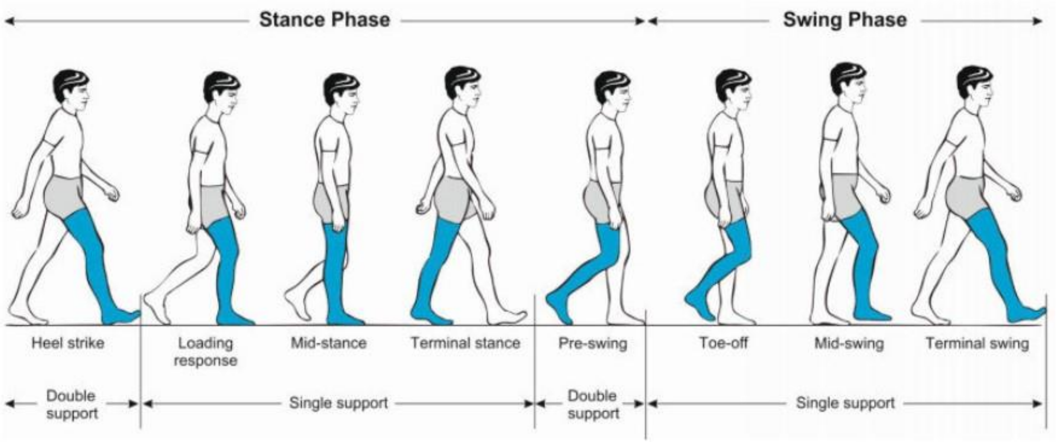


*Figure 2.1.2.4: Gait cycle sub-phases. Picture from [8].*

## 2.2 Gait analysis

Gait analysis is a systematic way of identifying any variations in the gait pattern and trying to find out the reasons associated with it and how they can affect the human walking [6]. The goal of gait analysis is the quantification of factors that control the functionality of lower limbs.

### 2.2.1 Gait parameters

To study gait in detail, a quantitative analysis can be performed, providing information about spatio-temporal parameters, kinematics, and dynamics of walking during the gait cycle. For the purposes of this thesis, the focus will be on spatio-temporal parameters and kinematic parameters.

#### 2.2.1.1 Spatio-temporal parameters of gait

Spatio-temporal parameters exploit gait repeatability, as they are related to phases of the gait cycle that are supposed to occur repeatedly. These parameters are essential for quantitatively assessing an individual's motor function as they provide a numerical description of the key events in gait.

The gait parameters can be divided into temporal and spatial [9]. A brief description of the main parameters is reported below.

Spatial parameters include:

- Stride length: Distance between successive points of IC of the same foot.
- Step length: Distance between successive points of IC of the one foot and the opposite foot.
- Step width: Lateral distance between the heels of the left and right feet. It is typically measured from the midpoint of one foot to the midpoint of the opposite foot as they make contact with the ground.

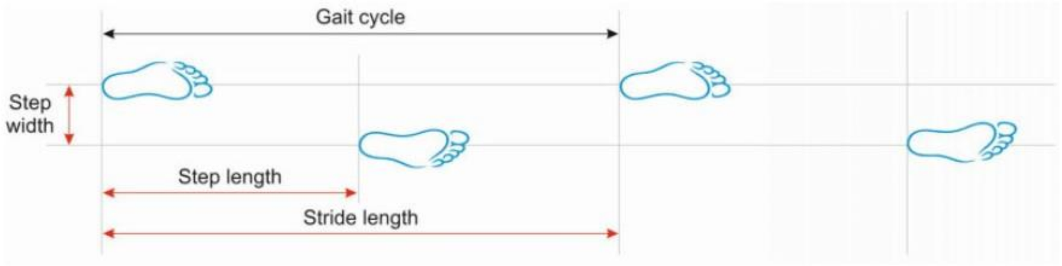


*Figure 2.2.1.1.1: Representation of spatial gait parameters. Picture from [8].*

Temporal parameters include:

- Stride time: Duration of the stride.
- Step time: Duration of the step.
- Cadence: The number of steps per unit time.

### 2.2.1.2 Joint kinematics

Joint kinematics focus on the study of the relative motion between two consecutive segments of the human body. This includes analysing the various components of joint motion, including direction, range, and quality of movement. In gait analysis, joint kinematics helps assessing how joints work during different activities.

The Standardization and Terminology Committee of the International Society of Biomechanics recommends the use of the Joint Coordinates System (JCS) proposed by Grood and Suntay in 1983 for the knee [10]. There are two guidelines that provide standard definitions for the JCS specifically for the shoulder, elbow, wrist, and hand [11] and for the hip, ankle, and spine [12]. According to [10], the JCS is defined starting from two cartesian coordinate systems (CCS) which are two body-fixed axes, $e_1$ and $e_3$, and one “floating” axis $e_2$, which is mutually perpendicular to them.

- **Shoulder joint**: Connects the clavicle, the scapula, and the humerus and consists of several joints, including the glenohumeral joint (GH), scapulothoracic (ST) joint, acromioclavicular joint (AC), and sternoclavicular joint (SC).

- JCS and motion for the SC joint (Figure 2.2.1.2.1): the $e_1$ and $e_3$ are fixed to the thorax and to the clavicle, respectively. These two body-fixed axes coincide with the Y-axis of the thorax CCS and the Z-axis of the clavicle CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of retraction or protraction; the body-fixed axis $e_3$ corresponds to the motion of rotation of the clavicle, and the floating axis $e_2$ corresponds to the motion elevation or depression of the clavicle.

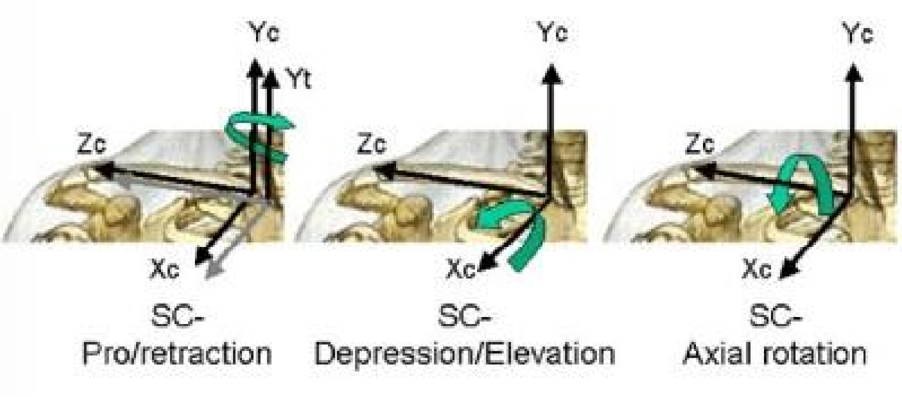


*Figure 2.2.1.2.1: Clavicle coordinate system and definition of SC motions.*

- JCS and motion for the AC joint (Figure 2.2.1.2.2): the $e_1$ and $e_3$ are fixed to the clavicle and to the scapula, respectively. The two body-fixed axes coincide with the Y-axis of the clavicle CCS and the Z-axis of the scapula CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of retraction or protraction of the AC; the body-fixed axis $e_3$ corresponds to the motion of rotation of the AC, and the floating axis $e_2$ corresponds to the motion AC-lateral or AC-medial rotation.

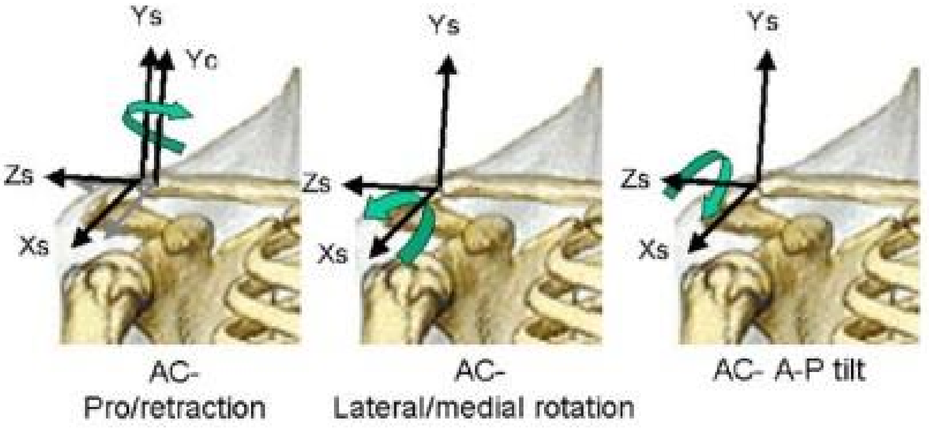


*Figure 2.2.1.2.2: Scapula coordinate system and definition of AC motion.*

- JSC and motion for the GH joint (Figure 2.2.1.2.3): Since $e_1$ and $e_3$ start in the same direction, the standard Grood and Suntay (floating axis) equations cannot be used. Instead, a Euler decomposition is used to find the corresponding angles [11]. The $e_1$ and $e_2$ are fixed to the scapula and to the humerus, respectively. The two body-fixed axes coincide with the Y-axis of the scapula CCS and the X-axis of the humerus CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of flexion/extension of the GH; the body-fixed axis $e_2$ corresponds to the motion of negative elevation (abduction/adduction) of the GH, and the axis $e_3$ corresponds to the GH-axial rotation, endo- or internal-rotation and exo- or external-rotation.

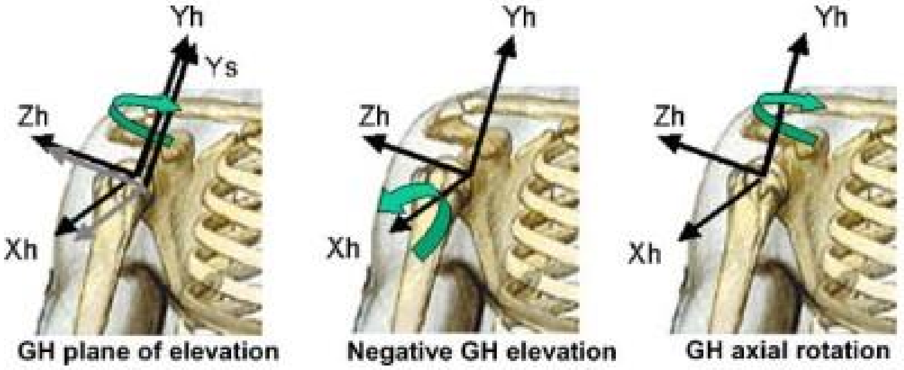


*Figure 2.2.1.2.3: Humerus coordinate system and definition of the GH motions.*

- JSC and motion for the ST joint: the $e_1$ and $e_3$ are fixed to the thorax and to the scapula, respectively. The two body-fixed axes coincide with the Y-axis of the thorax CCS and the Z-axis of the scapula CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of retraction or protraction of the scapula; the body-fixed axis $e_3$ corresponds to the motion of anterior of posterior rotation of the scapula, and the floating axis $e_2$ corresponds to the lateral or medial rotation of the scapula.

- **Elbow joint**: Connects the arm (humerus) and the forearm (radio and ulna), and consists of 3 joints, including the GH joint, the humeroulnar joint, and the radioulnar joint.
  - JCS and motion for the GH joint: the $e_1$ and $e_3$ are fixed to the humerus and to the forearm, respectively. The two body-fixed axes coincide with the Z-axis of the humerus CCS and the Y-axis of the forearm CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of flexion and hyperextension of the GH; the body-fixed axis $e_3$ corresponds to the motion of axial rotation of the forearm (pronation and supination), and the floating axis $e_2$ corresponds to the rotated X-axis of the forearm CCS and its motion is a passive response to the elbow flexion/extension.
  - JCS and motion for the humeroulnar joint: the $e_1$ and $e_3$ are fixed to the proximal segment (humerus) and to the distal segment (ulna), respectively. The two body-fixed axes coincide with the Z-axis of the humerus CCS and the Y-axis of the ulnar CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of flexion and hyperextension of the humerus; the body-fixed axis $e_3$ corresponds to the motion of axial rotation of the ulna, and the floating axis $e_2$ forms the so call carrying angle, the angle between the longitudinal axis of the ulna and the plane perpendicular to the flexion/extension axis.

  - JCS and motion for the radioulnar joint: the $e_1$ and $e_3$ are fixed to the proximal segment (ulna) and to the distal segment (radio), respectively. The axis $e_2$ is the rotated Z-axis of the radius CCS. The two body-fixed axes coincide with the X-axis of the ulnar CCS and the Y-axis of the radius CCS, respectively. The body-fixed axis $e_1$ corresponds to the orientation of the pro/supination axis relative to the ulna; the body-fixed axis $e_3$ corresponds to the pro/supination of the radius with respect to the ulna, and the floating axis $e_2$ corresponds to the abduction/adduction of the radius.

- **Wrist joint**: Connects the distal end of the radius and the carpal bones. It includes the midcarpal joint, which is the articulation between the proximal and distal rows of the carpal bones, the carpometacarpal joints, and the intercarpal joints.
  - JCS and motion for the interphalangeal, metacarpophalangeal, intercarpal, radiocarpal, and carpometacarpal joints: The $e_1$ and $e_3$ are fixed to the proximal (radius) and distal segment (carpal bones), respectively. The two body-fixed axes coincide with the Z-axis of the proximal segment CCS and the Y-axis of the distal segment CCS, respectively. The flexion/extension motion occurs on the body-fixed axis $e_1$, while the pronation/supination motion occurs on the body-fixed axis $e_3$. The floating axis $e_2$ corresponds to the abduction/adduction motion, or radial or ulnar deviation.
  - JCS and motion for the radioulnar joint: The radius and ulna may diverge slightly in the neutral posture, known as neutral forearm rotation. With the elbow flexed at 90 degrees, this position aligns the thumb with the shoulder. In the standard anatomical position, the radius is supinated relative to the ulna. An intermediate coordinate system is proposed by [11]. This system's

origin is at the radius's origin and aligns with the ulnar CCS in neutral forearm rotation. It is used to describe the radius's motion relative to the ulna. The $e_1$ and $e_3$ are fixed to the ulna and to the intermediate radial CCS, respectively. The two body-fixed axes coincide with the Y-axis of the intermediate radial CCS and the Z-axis of the intermediate radial CCS, respectively. The supination or pronation motion occurs on the body-fixed axis $e_1$, while the flexion/extension motion occurs on the body-fixed axis $e_3$. The floating axis $e_2$ corresponds to the radial-ulnar deviation.

- **Hip joint**: Connects the acetabulum of the pelvis and the head of femur. The $e_1$ and $e_3$ body-fixed axes of the JCS coincide with the Z-axis of the pelvic CCS and Y-axis of the right (or left) femur CCS, respectively. The body-fixed axis $e_1$ corresponds to the motion of flexion/extension, the body-fixed axis $e_3$ corresponds to the motion of internal-external rotation, and the floating axis $e_2$ corresponds to the abduction/adduction motion.

  The motion of flexion/extension occurs in the sagittal plane.
- **Knee joint**: Connects the femur and the tibia bones and consists in tibiofemoral and patellofemoral joints. The body-fixed axis $e_1$ coincides with the X-axis of the femoral CCS and corresponds to the motion of flexion-extension and it is also perpendicular to the femoral sagittal plane. The body-fixed axis $e_3$ coincides with the Z-axis of the tibial CCS and corresponds to the internal-external rotation of the tibia, and the floating axis $e_2$ corresponds to the adduction-abduction motion.

### 2.2.2 Instrumented measurements of gait parameters

Modern gait analysis heavily depends on the use of instrumented tools to obtain objective and accurate measurements of walking patterns and their variability [13].

Usually, different sensing modalities are used to collect the gait data which are categorized into two groups: vision-based and sensor-based [14]. Vision-based techniques for gait analysis utilize cameras to gather gait data, with cameras positioned at appropriate heights. These methods can be divided into two categories: marker-based and marker-less approaches. On the other hand, sensor-based techniques use two different types of sensing modalities to collect the gait data: wearable sensors and non-wearable sensors (Figure 2.2.2.1). Wearable systems use body-worn sensors, while non-wearable systems prioritize comfort of the individual and often employ force sensors installed on the floor.

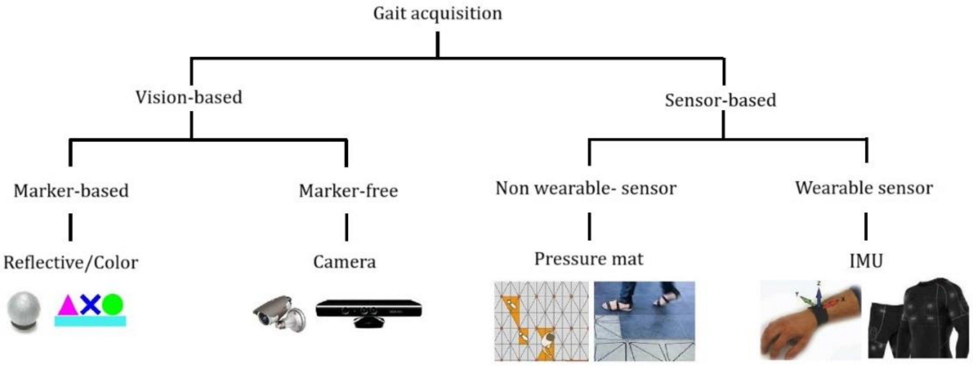


*Figure 2.2.2.1: Overview of different instrumented tools for gait analysis by [14].*

In this project a vision-based system with a marker-less approach is used to perform gait recognition (see Data collection).

#### 2.2.2.1.1 Video-based systems

Video-based systems are widely used in gait analysis due to their non-invasive nature and flexibility in capturing human movement. These systems typically use cameras to record a person's gait, which is then analysed to extract key features, such as spatio-temporal parameters and joint angles. They work by either tracking specific markers placed on the body, like reflective tapes or Light-Emitting Diode (LED) markers, or by using marker-less approaches that rely on image features such as shape, colour, and motion to identify and analyse gait patterns. The former are the opto-electronic systems (Figure 2.2.2.1.1), also referred to as optical Motion Capture (MOCAP) systems or stereophotogrammetry, in which the subject's motion is recorded by a set of visible or infrared (IR) light cameras; then, the resulting 2D digital images are processed to obtain 3D trajectories of markers placed on the subject's body [15]. The optical systems employed can be either passive, such as Vicon, or active, like Optotrak. Passive markers generally consist of spherical balls coated with a retroreflective material that reflects IR light from MOCAP system cameras. This use of IR light minimizes interference from visible light thanks to an IR pass filter placed over the camera lens. On the other hand, active markers are equipped with infrared LEDs located on body segments. Only one LED is illuminated at a time, which makes the identification of markers in the images more accurate. Due to their accuracy and reliability, stereophotogrammetry is considered as the gold standard (GS) for gait analysis; nevertheless, optical motion tracking systems are expensive, they need specialized laboratories, long calibration times, and long patient preparation.

The alternative is the marker-less systems, which use a standard video recording to capture movement without requiring markers to identify body positions and orientations, making them suitable also for surveillance applications, unlike marker-based systems. One of the most challenging and difficult steps is the video acquisition and representation of human movement. For these reasons, specific systems are required to capture the images. Nowadays, two main marker-less systems are used in MOCAP. The first one uses standard video cameras, while the second system relies on depth cameras, which use technologies like stereo vision or time-of-flight to capture both RGB and depth data. Both systems can be implemented using either single camera or multi-cameras for data collection. Once data are collected, pose estimation algorithms are used to detect and extract joint centres from images. These algorithms often employ deep learning techniques and are trained on large datasets with manually labelled key points [16]. These algorithms perform mathematical calculations on each image using, for instance, Convolutional Neural Networks (CNNs). A CNN is composed of multiple layers where each layer's output is the next layer's input. This architecture allows the network to progressively refine and interpret the image data to accurately estimate body poses. Prominent pose estimation algorithms include OpenPose and MoveNet [17].

There are many cameras on the market, i.e. Azure Kinect DK, Vicon Nexus, Intel RealSense D435 that differ in terms of acquisition and reconstruction of the image.

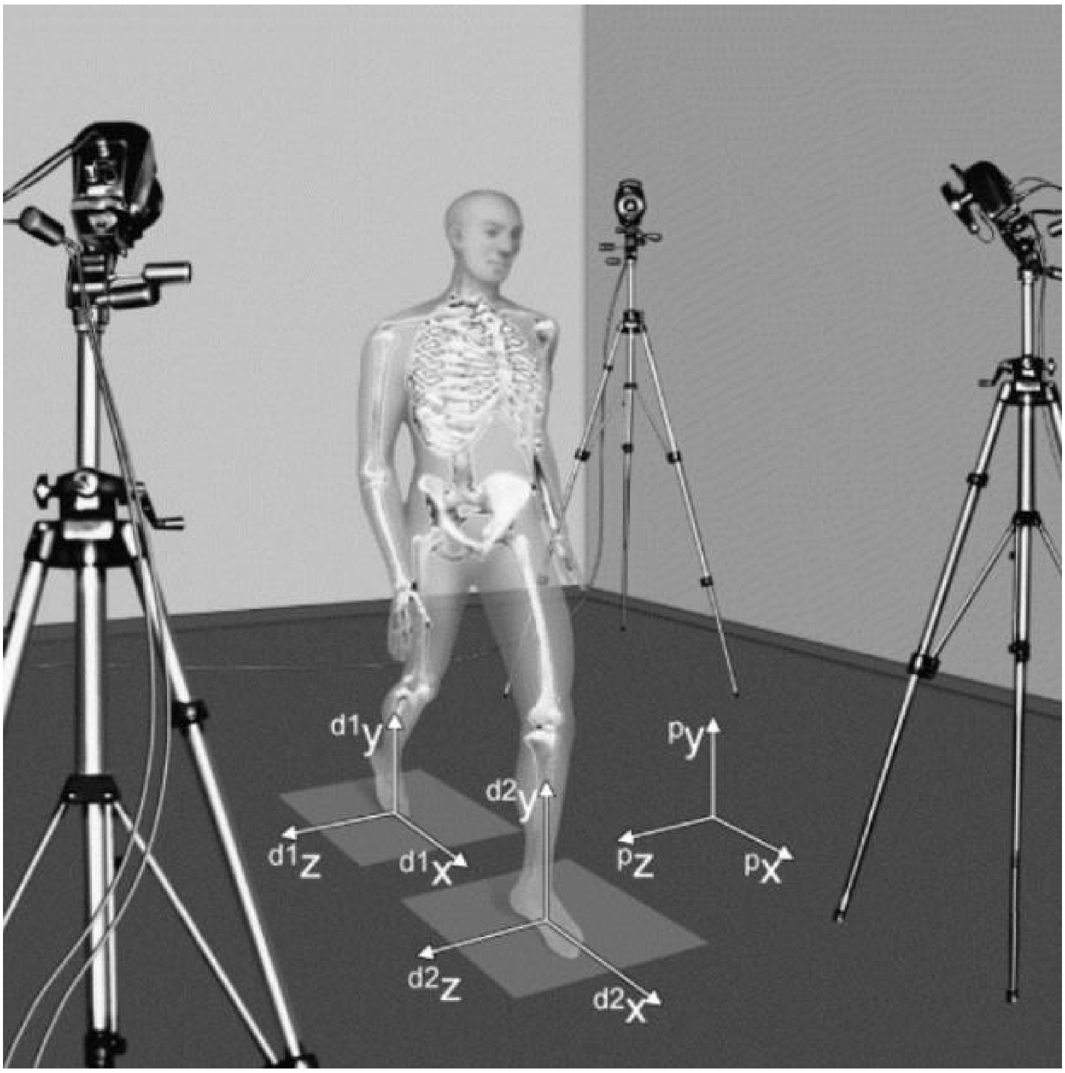


*Figure 2.2.2.1.1: Illustration of an opto-electronic system [15].*

#### 2.2.2.1.2 Other technologies for gait analysis

Non-optical systems include a large variety of instrumentations, in general based on inertial, electromagnetic, electromechanical or acoustic principles. As said before, sensor-based technologies use two types of sensing modalities to capture gait data: wearable sensors and non-wearable sensors.

Below is a brief overview of some of the sensor-based systems currently in use.

- **Force platforms:** Devices that detect the ground reaction force (GRF) beneath the foot in three directions, vertical, anterior-posterior, and medial-lateral, and deliver a current or voltage proportional to the detected pressure

[18] (Figure 2.2.2.2.1). They provide detailed information about the direction and point of application of the resultant reaction force. They are typically rectangular-shaped and sized between 40 cm and 60 cm and can provide accurate gait temporal parameters such as ICs.

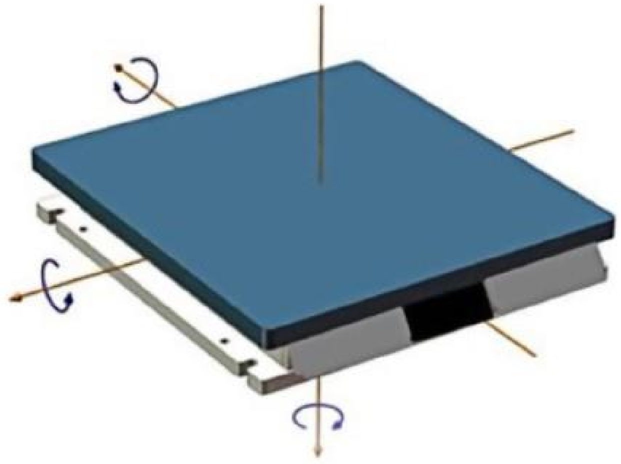


*Figure 2.2.2.2.1: 3D force platform with the direction of the measurable forces and moments [19].*

- **Pressure insoles:** Specialized footwear components designed to measure and analyse pressure distribution across the soles of the feet (Figure 2.2.2.2.2). These insoles are equipped with sensors that capture various data related to the forces exerted on different areas of the foot while walking, running, or standing. Sensors are typically embedded in a flexible plastic substrate to adhere to the foot shape. Pressure insoles can collect real-time data while the user is moving or standing.

*Figure 2.2.2.2.2: Illustration of a pressure insole with 16 sensing elements [20].*

- **Instrumented mat:** Device used to measure and analyse various aspects of foot pressure and gait dynamics. This mat is embedded with sensors and electronics to capture detailed data about the forces and pressure distribution exerted by the feet on different areas of the mat’s surface. They offer the possibility of measuring temporal and spatial parameters, but they are very bulky and not easily portable, so that they cannot be used in outside laboratory settings.
- **Magneto Inertial Measurement Units (MIMUs):** wearable devices composed by orthogonally mounted tri-axis sensors: a gyroscope which measures the angular rate, an accelerometer which measures the linear and gravity acceleration, a magnetometer which measures the magnetic field. They are small and lightweight, making them easy to attach to various parts of the body. Moreover, they can transmit data wirelessly to a computer or mobile device for analysis, allowing for real-time monitoring and feedback. The gyroscope suffers from drift over time and the magnetometer is highly

affected by the ferromagnetic distortions due to the Earth’s magnetic field [21].

### 2.2.3 Role of machine learning in gait studies

Machine learning (ML), a branch of artificial intelligence, involves developing algorithms that can learn from data and make predictions or decisions without being explicitly programmed. ML algorithms can be used to classify and recognize individuals based on their unique walking patterns. Specifically, in gait analysis ML is employed to model a biomechanical system by establishing the relationship between input data and outputs [22]. Typically, data pre-processing is necessary. The input data consists of raw, multidimensional arrays of size defined by the number of observations × the number of data features. The output of the model suggests a classification of gait events, activities, or disorders. ML evaluation models use an iterative mechanism by means of which the selection of input data is divided into training (TRS), validation (VS), and test sets (TS). The model is trained using a TRS and validated on a VS to prevent overfitting and perform hyperparameters tuning. Finally, the model is tested on unseen TS data, which was not used during the training phase. If the model achieves the desired accuracy, the process concludes. If not, the model's hyperparameters are adjusted, and the model is re-trained and re-evaluated iteratively until a suitable accuracy is reached.

The most used techniques include supervised learning, unsupervised
learning, and reinforcement learning [22].

- **Supervised learning:** This learning approach requires labelled data. The input data is an array of features vector, also called *predictors*, associated to their corresponding label. During training, the learning algorithm infers a function by mapping the training data instance to its corresponding label [23]. During training, the model adjusts its parameters to minimize the difference between its predictions and the actual labels. Such purpose is typically referred to as regression, for continuous outputs, or classification, for discrete outputs. In the framework of gait studies, many supervised algorithms have been explored:

I. Support Vector Machine (SVM): SVM has a good potential for generalization, even for small datasets, and stability. SVM is based on the principle of optimally separating data using hyperplanes and can efficiently perform classification tasks even when the data is not linearly separable by using kernels. Kernels transform the original feature space into a higher-dimensional space. Although it was originally designed to binary classification problems, it can be adapted to multi-class classification as well [14].

II. k-Nearest Neighbor (k-NN): The k-NN is the simplest and easy to implement classification tool which estimates the distance between the testing instance and picks the nearest k training vectors for average computation. Despite their simplicity in implementation, k-NNs are computationally expensive in classifying a test instance when the training size is large as it requires the estimation of the distance between k-nearest neighbors [14].

III. Decision Tree (DT): DT is a structure that is used to divide a data set containing many records into smaller sets by applying a series of decision rules [23], however, they usually fail when the degree of task complexity becomes too critical.

IV. Random forest (RF): RF is an ensemble of DTs; it builds multiple DTs and combines their results to improve overall performance. Each tree in the forest votes for a class, and the class with the most votes is chosen as the final prediction (Majority Voting).

V. Artificial Neural Networks (ANN): ANNs are inspired by the structure and function of the human brain. They are designed to recognize patterns, learn from data, and make decisions or predictions based on input data. Training of ANNs use different variants of the backpropagation algorithm [22].

VI. Naïve Bayes: This classification aims to determine the class of presented data to the system by a series of calculations defined according to the probability principles [23]. It is based on Bayes' Theorem with the “naïve” assumption that the features in a dataset are independent of each other, given the class label.

- **Unsupervised learning:** Also referred to as *clustering*, this learning process uses unlabelled data. The algorithm is expected to make self-discoveries designing the relation between various inputs to assess an output [22]. Most of the clustering techniques focus on the distance between all vector features. Such approaches were less explored in gait studies as it is a tedious job to properly identify the learning targets and manage a great amount of feature vectors. Therefore, where the relationship between different outcomes is ambiguous, certain techniques can be employed. For the use of large data sets, it is useful to combine classification with some methods for dimensional reduction [22].

- **Reinforcement learning:** This is the learning process based on data where the learning algorithm learns to act in an environment (training data) to maximize some notion of cumulative reward [23].

## 2.3 Gait recognition

Gait recognition, otherwise known as *gait signature*, is the recognition of some salient property, e.g., identity, style of walk, or pathology, based on the coordinated, cyclic motions that result in human locomotion [24]. The objective of gait recognition is to identify the discriminative features that differentiate people according to their style of walking. Gait has proven to be a well-recognized biometric modality, enabling the identification of individuals through their unique walking style [25]. It is important to stress that the way people walk is a strong correlate of their identity, with several studies confirming that both humans and machines can recognize individuals just by their gait [26]. The early studies of human gait in medical diagnosis and psychological analysis [27], [28], [29] reveal that human gait has 24 different components, making it unique to each individual when all factors are considered [30].

Gait recognition can be applied in several fields for different purposes:

- Surveillance and security: Used for monitoring individuals in a crowd, tracking suspects, and verifying identities for access control.
- Medical applications: In health care, gait recognition aids in diagnosing diseases related to gait such as Parkinson’s disease, orthopedic conditions, cerebral palsy, and stroke. In rehabilitation, it helps to monitor and analyse gait movement patterns of these patients such as stride length, step length, stance and swing phase to assess whether improvement has taken place.

- Biometrics: Used as a biometric identifier, similarly to fingerprints or facial recognition.

To recognize individuals, the gait recognition system must extract gait features from raw data collected during acquisitions (see Section 2.2.2), following a generic framework that consists of stages including data acquisition, feature representation, dimension reduction or feature selection, and classification [31] (Figure 2.3.1).

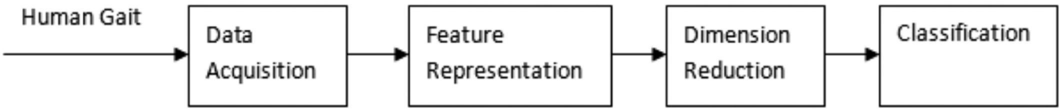


*Figure 2.3.1: Framework of gait recognition by [31].*

## 2.3.1 Current state of the art

Feature representation (Figure 2.3.1) is the first step for feature extraction. For video-based gait recognition, gait features are typically categorized into two main approaches: model-based approaches and model-free approaches (also known as appearance-based).

### 2.3.1.1 Model-based systems

In model-based systems, gait signatures are derived by modelling or tracking of body components (such as limbs, legs, arms, and thighs), which are employed for identification or verification of an individual. Model-based approaches can be divided into two categories: pose-based and sensor-based [18]. Both categories use a framework that relies on a stick-figure model to represent the human body. They use static and/or dynamic features that include, stride and cadence, distances of human body parts, length of body parts, height and joint angles between sets of rigid parts or from motion capture data. Model-based methods are easy to understand and are view-invariant, scale invariant and are not affected by background cluttering and noise, leading to more efficient representations.

However, these methods are usually more complex and computational expensive than model-free approaches, as it involves calculating key points in each frame.

The evolution of model-based gait recognition methods has been marked by several key contributions over the years. The first model-based feature representation approach is proposed by BenAbdelkader et al. [32], as they modelled the human body using two structural stride parameters, stride length and cadence, from the gait video for recognizing individuals. Bouchrika et al. [33] proposed a motion model using the elliptic Fourier descriptors to extract features from human joints and used them for person identification. To study and verify the importance of arm motion for recognition, Tafazzoli and Safabakhsh [34] proposed a model-based method for gait recognition based on the leg and arm movement. To create a model based on the movement of body parts, they employed active contour and Hough transform using anatomical facts. Yoo et al. [35] introduced a back-propagation neural network for automated gait recognition. Their method involved extracting nine body coordinates from silhouette images to form a 2D stick figure. Ten features were then derived from these figures and used by the neural network for recognition. To tackle view angle variation problem of 2D video-based gait recognition, Wang et al. [36] used second-generation Kinect V2 tool to create 3D skeleton-based gait database. Static and dynamic features were extracted for recognition of the person. Zhao et al. [37] proposed an approach based on 3D gait recognition. In this work 3D human model is created on video sequences taken from multiple cameras. Two feature sets are generated, static feature set including the length of key segments, and a dynamic feature set including motion trajectories of lower limb for recognition. In recent years, the ongoing advancements in model-based gait recognition reflect a trend towards integrating deep learning, multi-view approaches, and real-time processing to enhance recognition performance and applicability. Liao et al. [38] proposed a pose-based model-based gait recognition method in which they used a CNN model to extract features. Bari and Gavrilova [39] proposed a new architecture designed using deep learning neural network for a highly accurate and robust Kinect-based gait recognition.

#### 2.3.1.2 Model-free systems

In model-free gait recognition methods, no prior geometric model of the human body is formed. Instead, the focus is on the shape or movement characteristics of the human body's silhouettes. A silhouette is a solid shape and single black-coloured image of a person. Features are extracted directly from the binary outline of the gait, making the approach insensitive to colour and texture. The typical framework for a model-free approach involves several steps: detecting the subject in the captured video, removing the background using techniques like minimum or maximum filters, or histogram filtering, extracting the silhouette, and then applying classifiers for recognition [18]. This method is computationally less demanding compared to model-based approaches and is generally unaffected by the quality of the silhouette. However, while effective, model-free methods tend to be less robust to changes in viewpoint, appearance (such as different clothing, carried objects, or types of shoes), and scale.

In the last few decades, many model-free feature representation methods have been developed. Bobick and Davis [40] propose the motion-energy image (MEI) and motion-history image (MHI) to convert the temporal sequence of silhouettes to a 2D signal template. Han and Bhanu [41] employ the idea of MEI and propose the Gait Energy Image (GEI) for individual recognition for the first time. Liu and Zheng [42] introduced another gait representation method named Gait History Image

(GHI). Then, they compared GHI with GEI and several other temporal templates. Kale et al. [43] extracted the silhouette width of each image row from five key frames in a video sequence and modelled this with a Hidden Markov Model (HMM). Later, Kale et al. [27] combine the entire silhouette and the width of outer contour silhouette as gait features. Recently, Sheth et al. [44] proposed a CNN model with 8 layers to extract gait features from GEIs. Alsaggaf et al. [45] employed a Cycle Generative Adversarial Networks (CCGANs) approach to recognize individuals by their gait patterns without the need for additional support. Their method converts a distorted GEI caused by various factors into a normal GEI. In Guo et al. study [46], a Gabor filter is used to extract gait features from GEIs, since it can extract features of different directions and scales. A linear discriminant analysis (LDA) is used to tackle the problem that the feature dimension restricts the process. Finally, the improved local coupled extreme learning machine based on particle swarm optimization is used for the classification process of the extracted features of the gait.

This thesis proposes a model-based, marker-less method for gait recognition. The marker-less approach is chosen for its suitability in surveillance applications, while a model-based method is endorsed for being easy to understand, view- and scale-invariant, and less affected by background noise.

# 3 Data collection

This chapter describes the process of data collection, offering an explanation of the systems employed during the experiments. It includes a detailed account of subject selection, the setup, and the experimental protocol designed to capture the required data.

## 3.1 Experimental Setup

For the thesis purposes, the experimental setup (Figure 3.1.1) included:

- a green carpet, approximately 5 meters long and 1.45 meters wide.
- two Azure Kinect sensors (Microsoft Corporation), positioned laterally, about 2 meters from the midpoint line of the carpet, at a height of approximately 1.1 meters.
- one Vicon VUE camera (Vicon Motion Systems Ltd). The camera is placed 3 meters from the end of the carpet at a height of about 1 meter.

Both types of RGB cameras are sensitive to visible light, allowing them to function without the need for infrared markers. Here are some specifications:

- **Azure Kinect DK**: This device captures visual data using a 12MP RGB camera that has a frame rate of 30 frames per second, and a resolution of 1280 x 720. It is well-suited for capturing detailed visual information with accurate colour representation [47].
- **Vicon camera**: This device captures visual data using a 2.1MP RGB camera and has a higher frame rate of 60 frames per seconds, and a resolution of 1920 x 1080 [48].

*Figure 3.1.1: The experimental setup in the laboratory of the Polytechnic of Turin with a) Azure Kinect sensors, b) Vicon VUE camera, and c) green carpet.*

## 3.2 Experimental protocol

Fifteen healthy subjects (Table 3.2.1) were chosen to be studied in three different walking speed conditions.

| **Subject** | **Gender (M/F)** | **Age (y.o.)** | **Height (cm)** | **Weight (kg)** | **Shoe size (EU)** | **Waist width (cm)** |
|---|---|---|---|---|---|---|
| 0 | M | 24 | 179 | 70 | 46 | 32.5 |
| 1 | F | 26 | 173 | 58 | 39 | 26 |
| 2 | F | 28 | 170 | 70 | 39 | 26 |
| 3 | F | 24 | 165 | 52 | 39 | 28.5 |
| 4 | M | 33 | 180 | 77 | 43 | 31 |
| 5 | M | 22 | 187 | 74 | 44 | 29 |
| 6 | M | 26 | 184 | 77 | 43 | 29 |
| 7 | M | 26 | 178 | 73 | 43.5 | 29 |
| 8 | F | 26 | 172 | 62 | 40 | 25 |
| 9 | F | 26 | 155 | 49 | 36 | 23 |
| 10 | F | 25 | 165 | 58 | 36 | 22 |
| 11 | M | 23 | 176 | 79 | 42 | 30 |
| 12 | F | 23 | 168 | 57 | 38 | 25 |
| 13 | F | 26 | 170 | 63 | 39 | 26 |
| 14 | M | 30 | 180 | 80 | 45 | 27 |
| **Mean** | 53% (F) | 25,9 | 173,5 | 66,6 | 40,6 | 26,8 |
| **Std. dev.** | - | 2,8 | 8,4 | 10,2 | 3,2 | 2,7 |

*Table 3.2.1: Summary of participants recruited for acquisitions.*

Each subject performed four tasks, called *Tests*. Each test is composed of ten repetitions, called *Trials*. Hereafter each test is briefly described.

1. **Standing (Test 0):** Before starting the trials, each subject was asked to stand still for at least 2s in front of each one of the cameras. During the acquisition, the subject was asked to keep his arms along his hips. Such Test allows the extraction of the anthropometric parameters for each subject.
2. **Walk at comfortable speed (Test 1):** The participant walked at his own comfortable pace along a straight 5-meter path. Such Test is repeated 10 times (Trials).
3. **Walk at slower speed (Test 2):** The participant walked at a slower pace than his comfortable pace along a straight 5-meter path. Such Test is repeated 10 times (Trials).
4. **Walk at faster speed (Test 3):** The participant walked as fast as he could along a straight 5-meter path. Such Test is repeated 10 times (Trials).

Participants were requested to wear shorts and a skinny T-shirt, and they were instructed to walk along the midpoint line of the green carpet towards the Vicon RGB camera.

# 4 Methods

As mentioned in the previous chapters, the aim of this thesis was to develop a biomechanically driven marker-less method based on multiple RGB cameras for recognizing people by their gait. In particular, the proposed method consists of several phases as illustrated in the block diagram below (Figure 4.1).

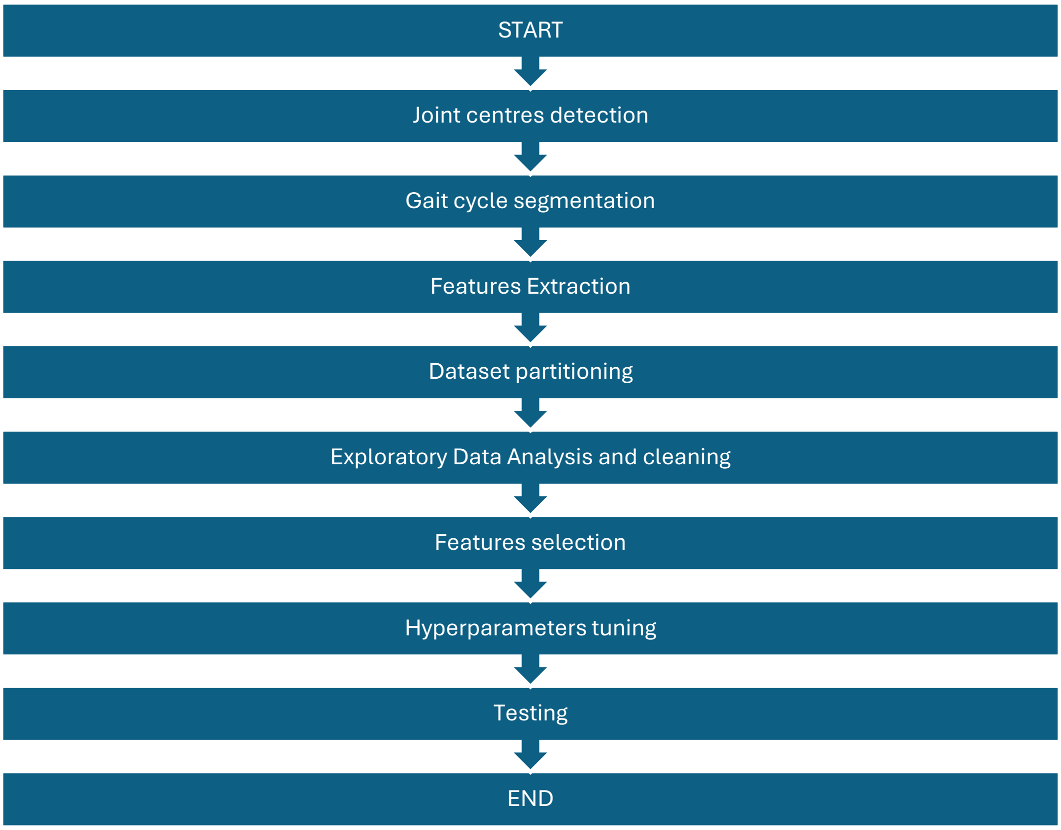


*Figure 4.1: General overview of the proposed method for people recognition by their gait.*

All the steps of the method were performed using MATLAB (MATLAB version R2022b. Natick, MA: The MathWorks, Inc., 2022) except for joint centres detection, which was performed using Python (version 3.11).

## 4.1 MoveNet

Developed by Google in Python in 2021, MoveNet is a bottom-up estimation model, which means that it first detects the human joints of all persons, and then assemble these joints into poses for each person using heatmaps [49]. Human joints are referred to as *keypoints*. The model was trained on healthy subjects using COCO dataset [50] and another Google's internal dataset called Active. COCO dataset is the standard benchmark dataset for detection, while Active was produced by labelling keypoints (using COCO's standard 17 keypoints) on fitness and dance videos from YouTube. The seventeen keypoints include the nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles on both sides of the body.

The MoveNet architecture (Figure 4.1.1) uses a bottom-up approach for human pose estimation, detecting 17 keypoints through heatmaps. Its backbone is MobileNetV2 combined with a Feature Pyramid Network for high-resolution outputs. The architecture includes four prediction heads: a person center heatmap, keypoint regression, keypoint heatmap, and a 2D offset field, all processed in parallel. Heatmaps contain confidence levels for each joint: once a person center heatmap is prepared to identify each person, the location with the highest score is selected [51]. MoveNet can identify up to six skeletons in an image, predicting seventeen keypoints for each skeleton.

MoveNet has two variants: Lightning and Thunder. The former can be used for applications that require speed and the latter for applications that require accuracy. The two models differ in input size and depth multiplier. In terms of input, Lightning receives a video or an image of a fixed size (192×192) and three channels and employs 1.0 depth multiplier. In contrast, Thunder receives an input of the size 256×256 and three channels and employs 1.75 depth multiplier. The depth multiplier changes the number of channels of the input video/image, which generally adopts the red-green-blue (RGB) format. Yet feature maps can also be regarded as one channel in each layer. Meanwhile, Thunder has 1.75 times more layers for deep learning than Lightning [52].

MoveNet is hosted on Tensorflow-Hub, an open repository of trained ML models exploitable for fine-tuning.

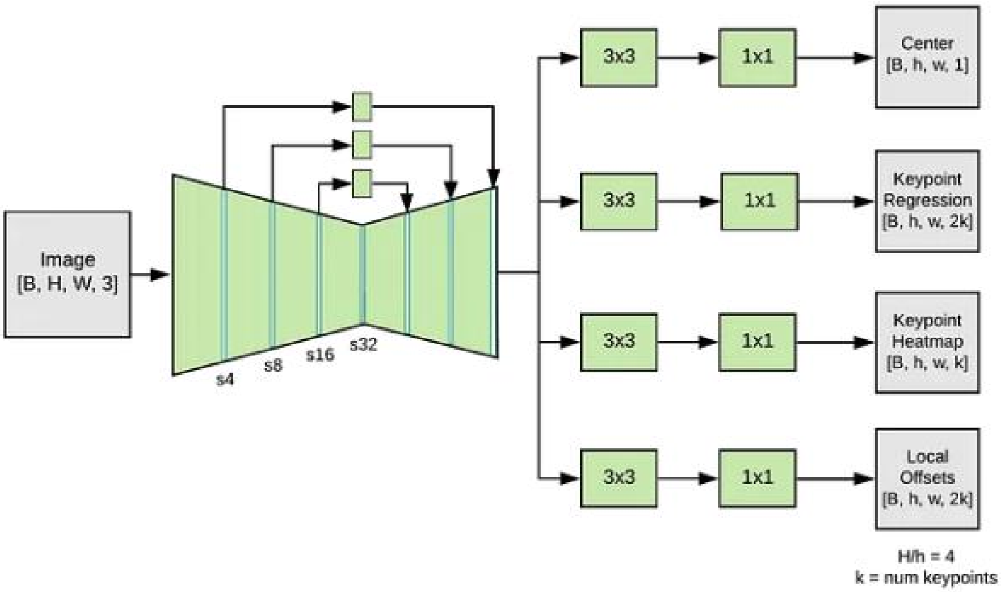


*Figure 4.1.1: MoveNet architecture from [51].*

## 4.2 Joint centres detection

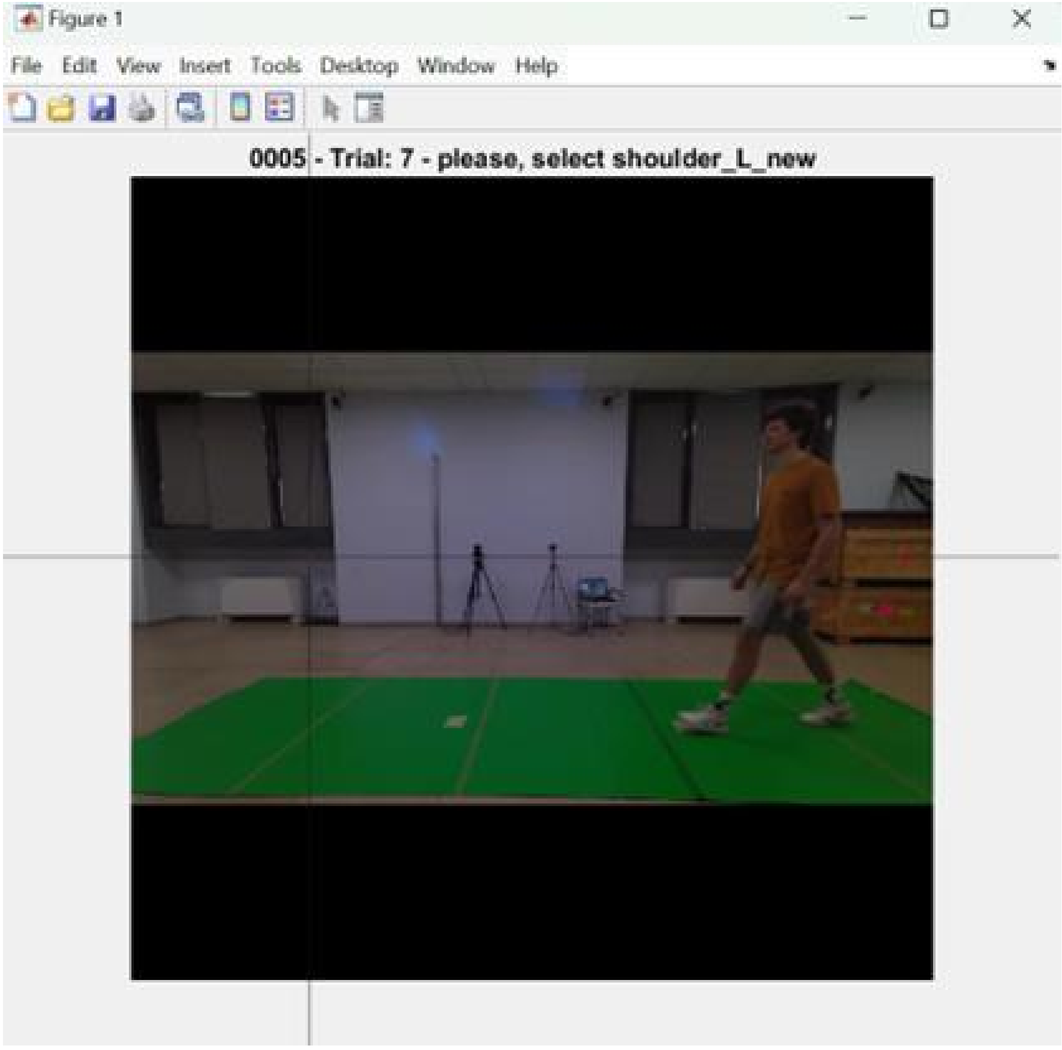
The first step of the proposed method involved the extraction of human keypoints. This required a pose estimation algorithm. Among the various available algorithms, MoveNet was chosen for its superior accuracy in identifying keypoints and ease of use. In this thesis, the Lightning variant was chosen for its efficiency [53].

To enhance precision, the method involved merging the six detected skeletons. The confidence levels for each keypoint across the six skeletons were analysed, and the keypoints with the highest confidence levels were selected. Once all keypoints were detected, the algorithm assembled these joints into complete poses for each person. The final output was a set of coordinates corresponding to the selected keypoints.

However, some limitations were noted during its application. Since MoveNet was trained specifically on the frontal plane, errors occurred with images acquired on the sagittal plane. Additionally, frontal plane acquisitions were sometimes affected by errors when the subject moved close the frontal RGB camera or due to laboratory lighting and objects in the background that could not be removed from the environment.

For this reason, manual labelling was performed. The first steps differ for sagittal and frontal plane. For the sagittal plane, before manually identifying wrong keypoints, the distance between adjacent keypoints was computed for each frame (e.g., the distance between shoulder and elbow). The average distance between adjacent keypoints was then calculated and each distance value was compared to the corresponding mean value ± the interquartile range. Keypoints that had a distance value higher than their corresponding average value were removed. Once all the distances have been examined, a linear interpolation was computed to handle the missing values.

An additional control was performed by plotting each keypoint individually to verify their accuracy (Figure 4.2.1). Corrections were applied exclusively to the keypoints on the same side as the acquisition plane.



0005 - Trial: 7 - please, select shoulder_L_new

*Figure 4.2.1: MATLAB interface for adjusting manually the wrong keypoint. In title there is the subject's ID, the trial number and which keypoint the user must adjust.*

New keypoint coordinates were organized in a MATLAB struct which was then saved as .json file.

For the frontal plane, where MoveNet performs more effectively, wrong keypoints were identified directly from the images provided by MoveNet. Keypoints were manually inserted and organized in a MATLAB struct which was then saved as .json file.

## 4.3 Gait cycle segmentation

Once all the keypoints were adjusted, the next step was to segment the gait cycle into individual strides by identifying two consecutive foot contacts with the ground.

The number of strides identified varied depending on the subject and on the speed at which the subject walked.

### 4.3.1 Sagittal plane

The heel strikes identification on the sagittal plane was based on the ankle’s coordinates captured during walking [54]. For this purpose, the derivative of the ankle coordinates was computed to capture the rate of change in position. A 4th order Butterworth filter was applied to the ankle derivatives to remove noise and smooth the data. The cutoff frequency for the filter was set to 5 Hz. If the data corresponded to the left side, the filtered derivative was inverted to standardize the detection process. A threshold was defined to detect the stance phase of the gait, that happened when the derivative was below this threshold. Continuous regions where the derivative remains below the threshold were identified, indicating the foot was in contact with the ground. ICs were extracted from the stance phases. These points corresponded to the frames where the heel strike occurred. Spurious initial contact points were removed based on the temporal distance between consecutive points to ensure only valid strides were identified. In case the last contact point was not automatically selected, it was manually identified.

### 4.3.2 Frontal plane

Heel strikes on the frontal plane were manually identified. The choice was to select the same heel contacts identified on the sagittal plane, ensuring consistency in the identification process across different planes.

## 4.4 Features for gait recognition

After segmenting the gait cycle, a comprehensive set of gait features was calculated for each stride of each trial. Specifically, 48 gait features were derived from the sagittal plane and 57 gait features were extracted from the frontal plane, resulting in a total of 105 gait features per stride for each trial. These features included time-domain features, cross-correlation between limbs, synchrony and asymmetry indices, and frequency-domain features. Some features were extracted both from the frontal and sagittal plane, while others were specific to one plane.

### 4.4.1 Time-domain features

According to [55] and [56], some of the most relevant biomechanical parameters used for gait analysis in a healthy adult population are spatio-temporal features and joint angles.

#### 4.4.1.1 Spatio-temporal parameters

Spatio-temporal parameters used in the present study included stride length, stride duration and stride velocity. These were calculated by taking the ICs identified through gait cycle segmentation (see Section 4.3.1) as input. The stride length was computed as the Euclidean distance between two consecutive ICs of the same foot. The stride duration was determined by dividing the number of frames occurring between two consecutive ICs of the same foot by the camera's sampling rate. Finally, the stride velocity was calculated as the stride length divided by the stride duration. Additionally, parameters were normalized by the subject's leg length to account for anthropometric differences between subjects.

ICs identification was crucial for determining the total number of strides per trial since each subsequent feature was computed stride-by-stride.

Step length was computed as the Euclidean distance between the ICs of the contralateral ankles, right to left for a right step and vice versa for a left step. Step duration was determined by the absolute difference between the frame numbers corresponding to these two contralateral ICs divided by the camera's sampling rate.

Again, parameters are normalized by the subject's leg length to account for anthropometric differences between subjects.

For calculating stride length, stride duration and stride velocity on the frontal plane, the same low-pass Butterworth filter is applied (see in Section 4.3.1) to the ankle and hip coordinates to remove high-frequency noise. The hip and ankle x-coordinates were normalized by the shoulder distance, while the y-coordinates were normalized by the subject's height. Depending on the IC side, stride lengths, durations, and velocities for both left and right strides using the filtered and normalized ankle coordinates and IC events are computed. Stride length was calculated as the Euclidean distance between the ankle positions at two consecutive IC events; stride duration was calculated as the difference of the frame numbers corresponding to the two consecutive IC events divided by the camera's sampling rate and stride velocity was calculated by dividing the stride length by the stride duration. Stride length and velocity were normalized by the subject's average leg length to account for individual differences in leg length.

### 4.4.1.2 Joint angles estimation

#### 4.4.1.2.1 Sagittal plane

On the sagittal plane, the head inclination, the trunk flexion, the hip and knee angles were estimated for each stride. For left strides, joint coordinates of the left side of acquisition were used, and vice versa for right strides.

The joint angle trajectories throughout the gait cycle were computed as follows:

- The head inclination angle was calculated as the supplementary angle relative to the angle formed by the SHOULDER-HIP normalized vector $a_1$ (trunk axis) and the EAR-SHOULDER normalized vector $a_2$ (head axis).

$$a_1 = \frac{SHOULDER - HIP}{|SHOULDER - HIP|} \tag{1}$$

$$a_2 = \frac{EAR - SHOULDER}{|EAR - SHOULDER|} \tag{2}$$

$$\theta_{head} = 180° - \cos^{-1}(-a_1 \cdot a_2) \tag{3}$$

- The trunk flexion angle was calculated as the inclination between the vertical axis $a_0 = [1\ 0\ 0]$ and the trunk axis.

$$\theta_{trunk} = \cos^{-1}(a_0 \cdot a_1) \tag{4}$$

The elbow angle was calculated as the inclination between the SHOULDER-ELBOW normalized vector $a_3$ (upper arm axis) and the WRIST-ELBOW normalized vector $a_4$ (forearm axis).

$$a_3 = \frac{SHOULDER - ELBOW}{|SHOULDER - ELBOW|} \tag{5}$$

$$a_4 = \frac{WRIST - ELBOW}{|WRIST - ELBOW|} \tag{6}$$

$$\theta_{elbow} = \cos^{-1}(a_4 \cdot a_3) \tag{7}$$

- The hip angle was calculated as 90 degrees minus the thigh angle. The thigh angle was computed as the inclination between the hip and the knee.

  $$\alpha_{thigh} = \tan^{-1}\left(-\frac{hip_y - knee_y}{hip_x - knee_x}\right) \tag{8}$$

  where $hip_y$ and $hip_x$ represent the y- and x-coordinate of the hip, respectively, and $knee_y$ and $knee_x$ represent the y- and x-coordinate of the knee, respectively.

  The thigh angle was converted from radiant to degree.

- The knee angle was calculated as the difference between the thigh and shank angles. The shank angle was computed as the inclination between the knee and the ankle.

  $$\alpha_{shank} = \tan^{-1}\left(-\frac{ankle_y - knee_y}{ankle_x - knee_x}\right) \tag{9}$$

  where $ankle_y$ and $ankle_x$ represent the y- and x-coordinate of the ankle, respectively, and $knee_y$ and $knee_x$ represent the y- and x-coordinate of the knee, respectively.

  After converting from radiant to degrees, the knee angle was computed:

  $$\theta_{knee} = \alpha_{thigh} - \alpha_{shank} \tag{10}$$

Each stride was made of a different number of frames, so the resulting angles were referred to the percentage of the gait cycle (0-100%) through spline interpolation.

The kinematic curve was then filtered by applying a 4th-order Butterworth filter with a cutoff frequency of 5 Hz. For each stride, the interquartile range (IQR) of each angle was calculated, as shown below.

$$IQR = Q3 - Q1 \tag{11}$$

where Q1 represents the first quartile (25th percentile) of the angles and Q3 represents the third quartile (75th percentile) of the angles.

Additionally, according to [57], the arm swing angle was another typical feature of human walking. This feature was computed processing the kinematic data of shoulder, wrist and hip to determine the swing angle, its velocity and sway characteristics over a specified number of frames.

First, the wrist, hip, and shoulder coordinates were adjusted relative to the shoulder's starting position. The wrist velocity was then calculated and its 95th percentile was determined for both x and y coordinates. To determine the anterior and posterior sway, the distance between the wrist and hip along the y-axis was calculated. This distance was then filtered to remove noise by applying a 4th order Butterworth filter with a cutoff frequency of 5 Hz.

Anterior and posterior swing phases were identified based on the sign of the distance: for right strides, anterior sway occurs when the distance is greater than zero, while posterior sway when the distance is less than zero. For left strides, the conditions were reversed. Indices marking the start and end of these phases were determined, allowing the sway angles at key points to be computed as the dot product between $a_1$ and $a_2$, which were the HIP-SHOULDER and WRIST-SHOULDER normalized vectors, respectively. The total arm swing angle over the gait cycle was then interpolated.

Next, the angular velocity of the swing angle was calculated and its 95th percentile was determined. The peaks of the anterior and posterior sway angles were identified and averaged. Finally, the total arm swing angle was calculated by summing the maximum anterior and posterior sway angles.

#### 4.4.1.2.2 Frontal plane

On the frontal plane, the head inclination, the trunk oscillation, the pelvis inclination and both the right and left elbow, shoulder, hip and knee angles were estimated for each stride. The joint angle trajectories throughout the gait cycle were computed as follows:

- The head inclination angle was calculated as the inclination between the trunk axis $a_{13}$ and the vector from the midpoint of the shoulders (MPSH) to the nose $a_4$. The trunk axis was defined as the vector between the MPSH and the midpoint of the hip (MPP).

$$a_4 = \frac{MPSH - NOSE}{|MPSH - NOSE|} \tag{12}$$

$$a_{13} = \frac{MPSH - MPP}{|MPSH - MPP|} \tag{13}$$

$$\theta_{head} = \cos^{-1}(a_{13} \cdot (-a_4)) \tag{14}$$

- The trunk oscillation angle was calculated as the inclination between the vertical axis [0 1 0] $a_2$ and trunk axis $a_{13}$.

$$\theta_{trunk} = \cos^{-1}(-a_{13} \cdot a_2) \tag{15}$$

- The right elbow angle was calculated as the inclination between the right upper arm and the right forearm.

  The upper arm was the RIGHT SHOULDER (RS) - RIGHT ELBOW (RE) normalized vector $a_3$ and the forearm was the RIGHT WRIST (RW) - RIGHT ELBOW (RE) normalized vector $a_8$.

$$a_3 = \frac{RS - RE}{|RS - RE|} \tag{16}$$

$$a_8 = \frac{RW - RE}{|RW - RE|} \tag{17}$$

$$\theta_{elbow_R} = \cos^{-1}(a_3 \cdot a_8) \tag{18}$$

- The right shoulder angle was calculated as the inclination between RS – LEFT SHOULDER (LS) normalized vector $a_{10}$ and the right upper arm $a_3$.

$$a_{10} = \frac{RS - LS}{|RS - LS|} \tag{19}$$

$$\theta_{shoulder_R} = \cos^{-1}(a_{10} \cdot a_3) \tag{20}$$

- The pelvis inclination angle was defined as the RIGHT HIP (RH) – LEFT HIP (LH) normalized vector $a_5$ and the horizontal axis [1 0 0] $a_6$.

$$a_5 = \frac{RH - LH}{|RH - LH|} \tag{21}$$

$$\theta_{pelvis} = \cos^{-1}(a_6 \cdot a_5) \tag{22}$$

- The right hip angle was calculated as the inclination between the LH – RH normalized vector $a_{11}$ and the RH – RIGHT KNEE (RK) normalized vector $a_{15}$.

$$a_{11} = \frac{LH - RH}{|LH - RH|} \tag{23}$$

$$a_{15} = \frac{RH - RK}{|RH - RK|} \tag{24}$$

$$\theta_{hip_R} = \cos^{-1}(a_{11} \cdot (-a_{15})) \tag{25}$$

- The right knee angle was calculated as the inclination between $a_{15}$ and the ANKLE R (ANKR) – RK normalized vector $a_{16}$.

$$a_{16} = \frac{ANKR - RK}{|ANKR - RK|} \tag{26}$$

$$\theta_{knee} = \cos^{-1}(a_{15} \cdot a_{16}) \tag{27}$$

The same computations were performed for the left-side angles.

Since each stride contained a different number of frames, the resulting angles were normalized to a percentage of the gait cycle (0-100%) using spline interpolation. Subsequently, the kinematic curve was filtered using a 4th order Butterworth filter with a cutoff frequency of 5 Hz. For each stride, the IQR of each angle was calculated, as shown in Equation (11).

#### 4.4.1.3 Cross-correlation between limbs

During gait, the movements of limbs are not independent but are synchronized in a unique, individual-specific manner. To capture this synchronization, cross-correlation analysis was performed between various limb angle pairs such as the shoulder and hip, shoulder and elbow, neck and hip, shoulder to shoulder, and hip to knee, both on sagittal and frontal plane. This was done by extracting three key parameters from the cross-correlation between these angles: the maximum value in the cross-correlation sequence and its corresponding index, the lag at which this maximum value occurred, and the value of the cross-correlation at zero lag, which was determined by indexing into the centre of the cross-correlation sequence.

#### 4.4.1.4 Synchrony and symmetry indices

According to [58], four coefficients are commonly used to measure symmetry: ratio index, symmetry index, gait asymmetry and symmetry angle.

In this thesis, the gait asymmetry (GA) index was used to evaluate symmetry between the steps. Specifically, on the sagittal plane, the GA was calculated between the right and left steps of both the acquisition sides, with the right side chosen for analysis due to its higher significance. The equation for GA is **GA**:

$$GA\ (\%) = \ln\left(\frac{X_R}{X_L}\right) * 100 \tag{28}$$

where $X_R$ and $X_L$ are the values of the right and left steps, respectively.

According to [57], the absolute symmetry angle (ASA) was used to evaluate the asymmetry between right and left limbs. ASA determines the asymmetry between right and left step lengths, step durations, and anterior sways.

The equation for ASA is (29):

$$ASA\ (\%) = \left|\frac{45° - \tan^{-1}\left(\frac{X_{MORE}}{X_{LESS}}\right)}{90°}\right| * 100 \tag{29}$$

where $X_{MORE}$ and $X_{LESS}$ are the parameter values associated with the major and minor step lengths, step durations, and anterior sways.

Considering the frontal plane, the synchrony between the right arm and the left leg and the left arm and the right leg was evaluated using the synchrony index (SI).

The SI was measured through the Pearson's correlation coefficient ρ [57] using the function `corrcoeff.m`. The formula for ρ is (30):

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{30}$$

where X and Y are the arm segment, calculated as the Euclidean distance between the right/left shoulder and right/left wrist, and the leg segment, calculated as the Euclidean distance between the left/right hip and the left/right ankle, respectively.

### 4.4.2 Frequency-domain features

To enhance discriminative power between subjects in gait recognition, specific frequency domain features were evaluated for different joints in both the sagittal and frontal planes. These features included the dominant frequency, peak amplitude, and full-width at half maximum (FWHM). The evaluation was conducted for both the x and y coordinates of the wrist in the sagittal plane and the head and pelvis in the frontal plane. Additionally, the wrist amplitude was evaluated in the frontal plane.

- Dominant Frequency: the primary frequency at which the movement occurs.
- Peak Amplitude: the maximum amplitude of the dominant frequency, indicating the intensity of the movement.
- Full-Width at Half Maximum (FWHM): the width of the frequency band at half of the peak amplitude, providing insight into the variability of the movement frequency.

First, the Power Spectral Density (PSD) of both x and y coordinate trajectories of the wrist, head, and pelvis was calculated using Welch's method (`pwelch.m`). A Hamming window (`hamming.m`) equal to the length of the trajectory was applied to reduce spectral leakage when performing the Fourier transform. The overlap was set to 0, meaning that no overlap was used between segments, and the number of FFT points (NFFT) was set to twice the length of the input trajectory to improve the frequency resolution in the PSD estimate. Then, the PSD and the corresponding frequency values were used to determine the dominant frequency, the peak amplitude, and the FWHM of the peak using the function `findpeaks.m`.

These features were evaluated on each trial.

On the sagittal plane, the wrist coordinates (both x and y) were referred to the shoulder coordinates. This transformation aligned the wrist movements with respect to the shoulder position, providing a more standardized measurement of wrist movement.

On the frontal plane, the coordinates of the head and pelvis were scaled with respect to the trunk axis, aligning these measurements along the body's central axis. Then, the x coordinate was normalized to the subject's shoulder distance: this normalization adjusts for variations in shoulder width between individuals, ensuring that the measurement of lateral movements was comparable across subjects. The y coordinate was normalized to the subject's height: this normalization accounts for differences in individual height, providing a standardized measurement of vertical displacement.

Similarly to the head and pelvis, first the right and left wrists were scaled with respect to the right and left shoulders, then the x and y coordinates of the wrists were normalized with respect to the subject's shoulder distance and height, respectively. This ensured that the amplitude of wrists movements was comparable across different body sizes and proportions.

Additionally, the attenuation coefficient in decibels (dB) between the IQR of the head and hip [59] was evaluated stride-by-stride, using the equation (31):

$$r = 20 * \log\left(\frac{IQR_{head}}{IQR_{hip}}\right) \tag{31}$$

## 4.5 Dataset partitioning

The following paragraph outlines the steps for the construction of the dataset. The data used for training and validating the models consisted of the features previously mentioned (see Section 4.4), extracted from 15 participants (see Table 3.2.1) during the acquisitions under three different walking conditions (see 3.2). To standardize data acquisitions and allow subjects to adjust to the test task, the first three trials of each test were rejected for every subject. The dataset was divided into a construction set (CS) and a test set (TS). The construction set was used to train and validate the models, while the test set was reserved for evaluating the trained model and assessing its generalization capabilities. For each walking condition, 5 trials are assigned to the CS, while 2 trials are assigned to the TS. By doing so, both the construction and test sets included data of all the participants in the dataset.

To ensure consistency and comparability, the construction set was normalized through z-score normalization, which involved transforming the data to have a mean of zero and a standard deviation of one. The z-score formula is (32):

$$\mathrm{z} = \frac{x - \mu}{\sigma} \tag{32}$$

where $x$ is the original feature value, $\mu$ is the mean of the feature values in the construction set, and $\sigma$ is the standard deviation of the feature values in the construction set. The test set was then normalized using the mean and standard deviation obtained from the construction set.

Furthermore, at each training stage, the construction set was further split according to a 5-folds cross-validation into a training set (TRS) for model training and a validation set (VS) to prevent data overfitting during training (Figure 4.5.1). Details about the partitioning of the CS will be provided in Section 4.7.1.

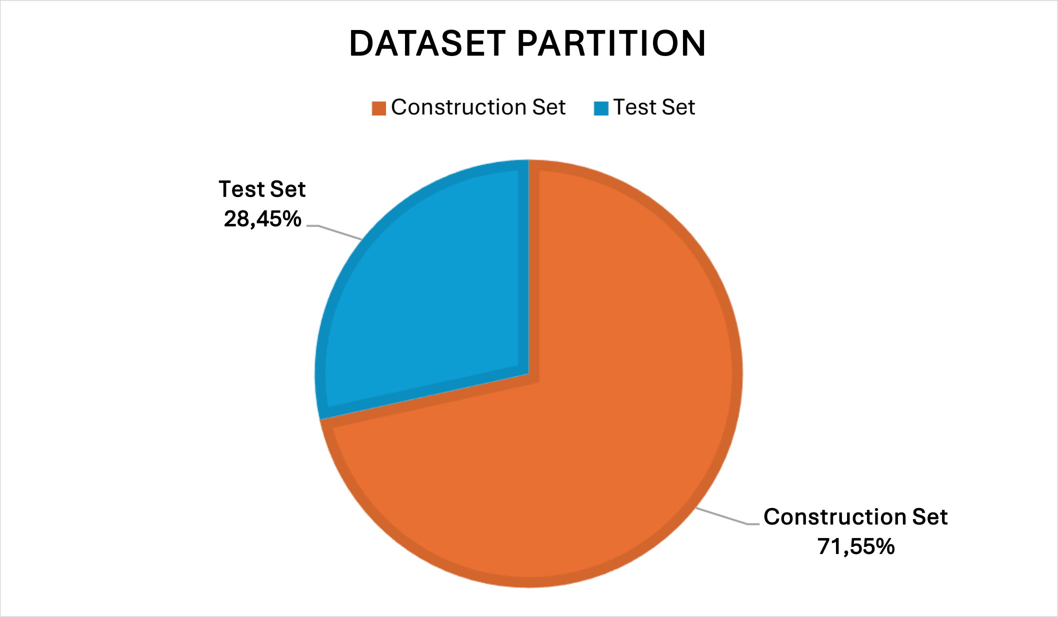


*Figure 4.5.1: The pie chart represents the partitioning of the dataset into CS and TS with their respective percentages relative to the entire dataset.*

## 4.6 Exploratory data analysis and cleaning

Exploratory data analysis was performed to visualize and analyse the final dataset before its partitioning. Initially, the dataset was visualized using boxplots and histograms to identify potential outliers (Figure 4.6.1). Following this, subject-specific outlier cleaning was conducted using MATLAB's Clean Outlier Data tool (Figure 4.6.2). This tool employs statistical techniques to identify and remove data points that deviate significantly from the rest of the dataset. It allows customization of both the cleaning and detection methods. In this case, the *fill outliers (nearest value)* cleaning method was used, where outliers were replaced with the nearest non-outlier value. The *quartiles* method, with a threshold of 0.5, was applied for outlier detection, identifying and adjusting data points that fall outside this specified range.

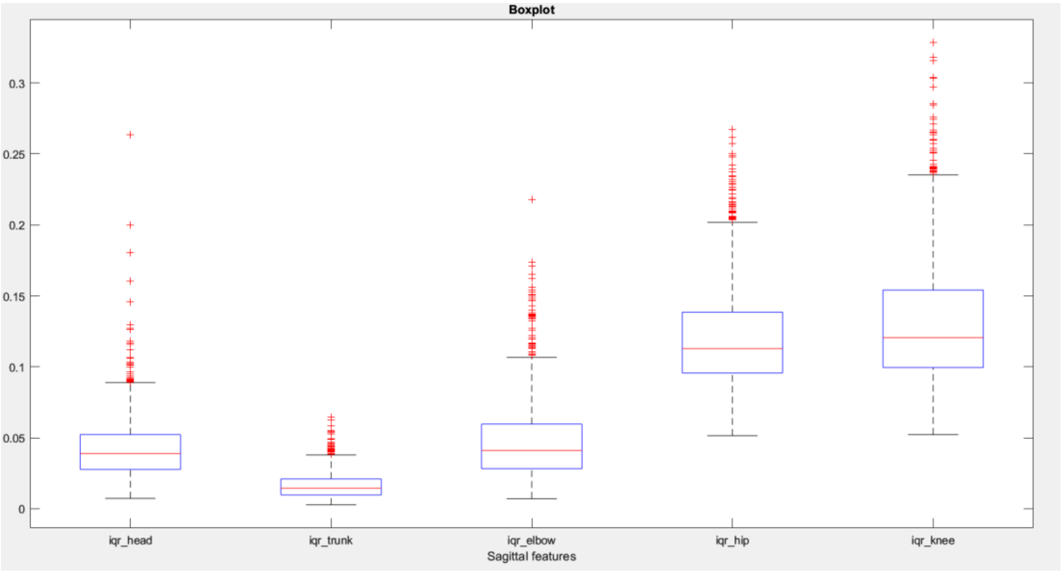
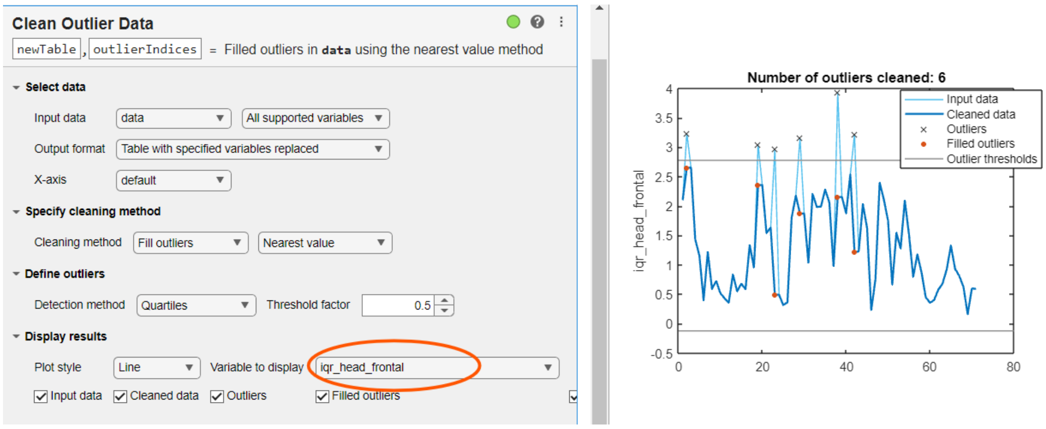


*Figure 4.6.1: Example of boxplot for some features of the sagittal plane.*



Clean Outlier Data

`newTable`, `outlierIndices` = Filled outliers in **data** using the nearest value method

Select data
Input data: data; All supported variables
Output format: Table with specified variables replaced
X-axis: default

Specify cleaning method
Cleaning method: Fill outliers; Nearest value

Define outliers
Detection method: Quartiles; Threshold factor: 0.5

Display results
Plot style: Line; Variable to display: iqr_head_frontal
☑ Input data ☑ Cleaned data ☑ Outliers ☑ Filled outliers

*Figure 4.6.2: MATLAB interface of the Clean Outlier Data tool on the left. The circle highlights where the user can select the variable to visualize and analyse. On the right, the plot of the variable (IQR of the head on the frontal plane) along with its identified outliers (x) and the values used to replace the outliers marked as orange dots. Grey lines represent the outlier thresholds of the detection method.*

Additionally, a manual review of all outliers for each subject was conducted to verify that they were correctly categorized as outliers due to acquisition or labelling errors. If it was determined that the value was influenced by the subject's walking style or speed, the value was not replaced.

## 4.7 Feature selection

Feature selection involves choosing a subset of features from the input data. In this thesis, due to the large number of features, identifying the most relevant ones was crucial for reducing the algorithm's computational complexity. This process can be done in two ways: forward (bottom-up) or backward (top-down). In forward selection, features are added one by one in an iterative manner until the optimal set of features is identified. In backward selection, the process starts with all features, and features are removed iteratively until the optimal subset is determined. For this work, a backward selection approach was used with a wrapper feature selection method. Wrapper feature selection is a method where different subsets of features are evaluated based on the performance of the model, in this case through a top-down approach that starts with the full set of features and iteratively removes the least relevant ones [60]. In particular, features were selected wrapping seven classification models: Decision Tree, Discriminant Analysis, Ensemble Classifier, Error-Correcting Output Codes (ECOC), k-Nearest Neighbors (k-NN), Naïve Bayes Classifier, and Neural Networks.

### 4.7.1 Feature selection protocol

Initially, the Minimum Redundancy Maximum Relevance (mRMR) algorithm was used to rank the features of the dataset [60]. The mRMR method prioritizes features that are highly relevant to the target variable (maximum relevance) while minimizing redundancy among the features (minimum redundancy). After features were ranked by the mRMR algorithm, a preliminary feature selection was performed by excluding features with mRMR score lower than 0.001, resulting in a subset of 19 features (Figure 4.7.1.1).

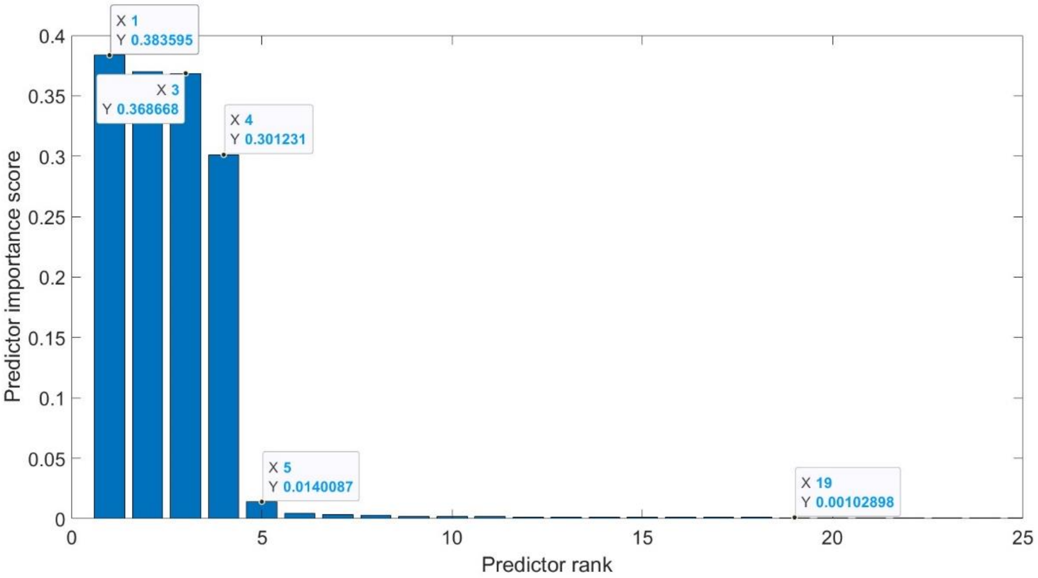


*Figure 4.7.1.1: The drop in score between features. Features with a score lower than 0.001 are excluded.*

Then, top-down wrapper feature selection was performed with the specified classification models (see Section 4.7).

At each iteration of the wrapper, cross-validation was performed to ensure robustness. The dataset was iteratively divided into training and validation sets, with 4 trials used for training and 1 trial for validation. The classification models were trained on data of the TRS, and their performance was assessed on the VS in terms of accuracy (see Section 4.9). Once completed all the five iterations of the cross-validation, the performance achieved by the model at that iteration of the wrapper was represented by the average accuracy on the VS. During each iteration of the wrapper, the feature set was adjusted by removing features. The algorithm stopped when the relative difference in accuracy between consecutive iterations was below a set tolerance of 0.001% or until the feature set included only one feature. For each model, the selected feature set was the one achieving the highest accuracy.

### 4.7.2 Selected features

Table 4.7.2.1 provides a list of the 19 features selected through the mRMR feature selection process, along with their reference plane and domain, and corresponding score.

| **Feature** | **Plane** | **Domain** | **Score** |
|---|---|---|---|
| Wrist amplitude (x coordinate) | Sagittal | Frequency | 0.3836 |
| IQR elbow R | Frontal | Time | 0.3702 |
| Wrist FWHM (x coordinate) | Sagittal | Frequency | 0.3687 |
| Cross correlation in 0 shoulder-shoulder | Sagittal | Time | 0.3012 |
| Wrist R amplitude (y coordinate) | Frontal | Frequency | 0.0140 |
| IQR head | Sagittal | Time | 0.0045 |
| 95th wrist R velocity (x coordinate) | Sagittal | Frequency | 0.0032 |
| Wrist L amplitude (x coordinate) | Frontal | Frequency | 0.0030 |
| Wrist L amplitude (y coordinate) | Frontal | Frequency | 0.0018 |
| Head max frequency (x coordinate) | Frontal | Frequency | 0.0018 |
| Cross correlation in 0 hip-knee | Sagittal | Time | 0.0017 |
| Wrist L velocity (x coordinate) | Sagittal | Frequency | 0.0015 |
| IQR elbow L | Frontal | Time | 0.0014 |
| ASA swing | Sagittal | Time | 0.0013 |
| Swing angle R | Sagittal | Time | 0.0012 |
| Wrist max frequency (y coordinate) | Sagittal | Frequency | 0.0012 |
| IQR shoulder L | Frontal | Time | 0.0012 |
| IQR elbow | Sagittal | Time | 0.0012 |

| Feature | Plane | Domain | Score |
|---|---|---|---|
| IQR knee | Sagittal | Time | 0.0010 |

*Table 4.7.2.1: List of the 19 selected features with their plane and score. L = left; R = right. Out of the 19 features, there is a predominance of features belonging to the sagittal plane (12 out of 19) and the time-domain (10 out of 19).*

The optimal subset of features could vary depending on the classification model used in the wrapper feature selection process. Initially, all models were trained with the complete set of 19 features. Through the wrapper feature selection, each model iteratively identified the most relevant subset of features, leading to different accuracies. Below, Table 4.7.2.2 shows the best feature set identified for each model, along with the corresponding accuracy achieved.

| Model | MATLAB function | Feature subset | Accuracy |
|---|---|---|---|
| Decision tree | `fitctree` | 1 | 43,77% |
| Discriminant Analysis Classifier | `fitcdiscr` | 1 | 61,53% |
| ECOC | `fitcecoc` | 1 | 67,43% |
| Ensemble classifier | `fitcensemble` | 2 | 56,06% |
| k-NN | `fitcknn` | 1 | 62,99% |
| Naive Bayes Classifier | `fitcnb` | 1 | 58,44% |
| Neural Networks | `fitcnet` | 1 | 55,84% |

*Table 4.7.2.2: List of the classification models with their MATLAB function used, the corresponding feature subset that achieved the highest accuracy, and the accuracy percentage. The feature subset is denoted as "1" if the highest accuracy was achieved using all 19 features. If fewer features were sufficient to achieve the best accuracy, the subset is labelled accordingly.*

For the feature selection stage, hyperparameters of the models were set empirically by choosing reasonable values, corresponding to the default parameters as reported in the documentation of the employed MATLAB routines (Table 4.7.2.2).

## 4.8 Hyperparameters tuning

Hyperparameters are the parameters set before the training process begins and remain unchanged during training. They can be automatically or manually adjusted during the tuning phase to enhance the model's performance.

After evaluating the performance of all models with default parameters, hyperparameters were tuned using a grid-search protocol combined with 5-fold cross-validation. Grid search is a methodical approach to hyperparameter tuning in machine learning models [61]. This method systematically explores multiple combinations of model parameters to identify the optimal set. It involves defining a parameter grid that outlines all possible parameter combinations to be tested and determining an appropriate method for evaluating model performance for each combination.

The hyperparameter tuning process utilized the feature subsets selected in the previous stage, ensuring that each model was optimized for performance based on the most relevant features identified during feature selection. In the following, a description of the hyperparameters tested for each model is provided [61], grouped by model type.

- **Decision Tree**:

  **Tested hyperparameters:**

  - *Split criterion*: **'gdi', 'twoing', 'deviance'**. Criteria that measure the quality of a split. Gdi stands for Gini's Diversity Index; twoing refers to the Twoing rule; deviance is often used in classification tasks as a measure of impurity.

- *Maximum Number of Splits*: **2, 4, 8, 16, 32, 64, 128**. This parameter controls the depth of the tree; a higher number of splits allows for more complex trees.
- *Minimum Leaf Size*: **from 2 to 10**. This parameter sets the smallest number of observations allowed in any leaf (final decision node) of the tree.

**Optimal hyperparameters configuration:**

- ➢ Maximum number of splits = 128
- ➢ Minimum leaf size = 4
- ➢ Criterion = 'deviance'

* **Discriminant Analysis**:

**Tested hyperparameters:**

- *Discriminant type*: **'linear', 'diaglinear'** (diagonal linear)**, 'pseudolinear', 'quadratic', 'diagquadratic'** (diagonal quadratic)**, 'pseudoquadratic'**. The discriminant type determines how the decision boundary is computed based on the distribution of classes.
- *Gamma*: **from 0 to 1**, values linearly spaced (only for 'linear' discriminant). It controls the regularization strength.

**Optimal hyperparameters configuration:**

- ➢ Discriminant type: 'pseudoquadratic'

* **ECOC**: ECOC models were implemented using the Support Vector Machine (SVM) template and categorized based on the kernel function (linear, radial basis function (RBF), polynomial) for ease of computation. Additionally, the Ensemble template was also implemented.

- **ECOC with Linear Kernel**

  **Tested hyperparameters:**

  - *Regularization Parameter*: **1, 10, 50**. This parameter controls the trade-off between achieving a low error on the training data and minimizing the model complexity to avoid overfitting.
  - *Coding Methods*: **'onevsone', 'denserandom', 'onevsall', 'ordinal', 'sparserandom'**. Coding methods determine how binary classifiers are combined to solve multiclass problems.

  **Optimal hyperparameters configuration**:

  - Regularization Parameter: 1
  - Coding Method: 'onevsone'

- **ECOC with RBF Kernel**

  **Tested hyperparameters:**

  - *Regularization Parameter*: **1, 10, 50**.
  - *Coding Methods*: **'onevsone', 'denserandom', 'onevsall', 'ordinal', 'sparserandom'**.
  - *Kernel scale*: **5, 10, 50, 100, 500, 1000**. The kernel scale affects the spread of the radial basis function (RBF), influencing the smoothness of the decision boundary.

  **Optimal hyperparameters configuration**:

  - Regularization Parameter: 10
  - Coding Method: 'onevsall'
  - Kernel scale: 5

- **ECOC with Polynomial Kernel**

  **Tested hyperparameters:**

  - *Regularization Parameter*: **1, 10, 50**.
  - *Coding Methods*: **'onevsone', 'denserandom', 'onevsall', 'ordinal', 'sparserandom'**.
  - *Polynomial degree*: ranged from **2 to 10**. This degree determines the flexibility of the decision boundary in polynomial kernel-based SVM models.

  **Optimal hyperparameters configuration**:

  - ➢ Regularization Parameter: 1
  - ➢ Coding Method: ‘sparserandom’
  - ➢ Polynomial degree: 2

- **ECOC with Ensemble Template**

  **Tested hyperparameters:**

  - *Methods*: **‘Bag’** (bagging)**, ‘AdaBoostM1’** (adaptive boosting).
  - *Number of learners*: **50, 100, 150, 200**. This refers to the number of base models (learners) in the ensemble.
  - *Coding Methods*: **'onevsone', 'denserandom', 'onevsall', 'ordinal', 'sparserandom'**.

  **Optimal hyperparameters configuration**:

  - ➢ Method: ‘Bag’
  - ➢ Number of Learners: 100
  - ➢ Coding Method: ‘denserandom’

* **Ensemble**:

**Tested hyperparameters:**

- *Aggregation methods*: **'Bag'** (bootstrap aggregating)**, 'AdaBoostM2'** (Adaptive Boosting).
- *Maximum Number of Splits*: **1, 71, 141, 211, 281, 351, 421, 491, 561, 631, 701**.
- *Learning rate*: **0.1**. This parameter determines the contribution of each model in the ensemble.

**Optimal hyperparameters configuration:**

- Aggregation method = 'Bag'
- Maximum number of splits = 141

* **k-NN**:

**Tested hyperparameters:**

- *k* (number of neighbors): ranged **from 1 to 30**. The value of *k* determines the number of nearest neighbors considered when classifying a new data point.
- *Distance metrics*: **'cityblock'** (Manhattan distance)**, 'chebychev'** (maximum coordinate difference)**, 'correlation'** (correlation-based distance)**, 'cosine', 'euclidean', 'hamming', 'jaccard', 'mahalanobis', 'minkowski', 'seuclidean', 'spearman'**. Different metrics to measure distance between data points.

**Optimal hyperparameters configuration:**

- k = 5
- Distance metric = 'cityblock', this distance sums the absolute differences across all dimensions.

* **Naïve Bayes Classifier**:

**Tested hyperparameters:**

- *Distributions*: **'kernel', 'mvmn'** (multinomial)**, 'normal'**.
- *Kernel functions*: **'box', 'epanechnikov', 'normal', 'triangle'** (only for 'kernel' distribution). These kernel functions determine the shape of the curve used to estimate the probability distribution of the data.

**Optimal hyperparameters configuration:**

- ➢ Distribution = 'normal', it assumes data follows a Gaussian distribution.

* **Neural Network**:

**Tested hyperparameters:**

- *Activation functions*: **'relu', 'tanh', 'sigmoid', 'none'**.
- *Hidden layers size*: **[10 0 0], [10 10 0], [10 10 10]**. The hidden layer configuration determines the complexity of the neural network model. The first configuration consists of one hidden layer of 10 neurons, the second configuration two hidden layers of 10 neurons each, and the third configuration three hidden layers of 10 neurons each.
- *Learning rate*: **from 0 to 423.73**, where these values are obtained by dividing the range from $10^{-5}$ to $10^{5}$ by the length of the construction set. This parameter controls how much the model's weights are adjusted with respect to the loss gradient.

**Optimal hyperparameters configuration:**

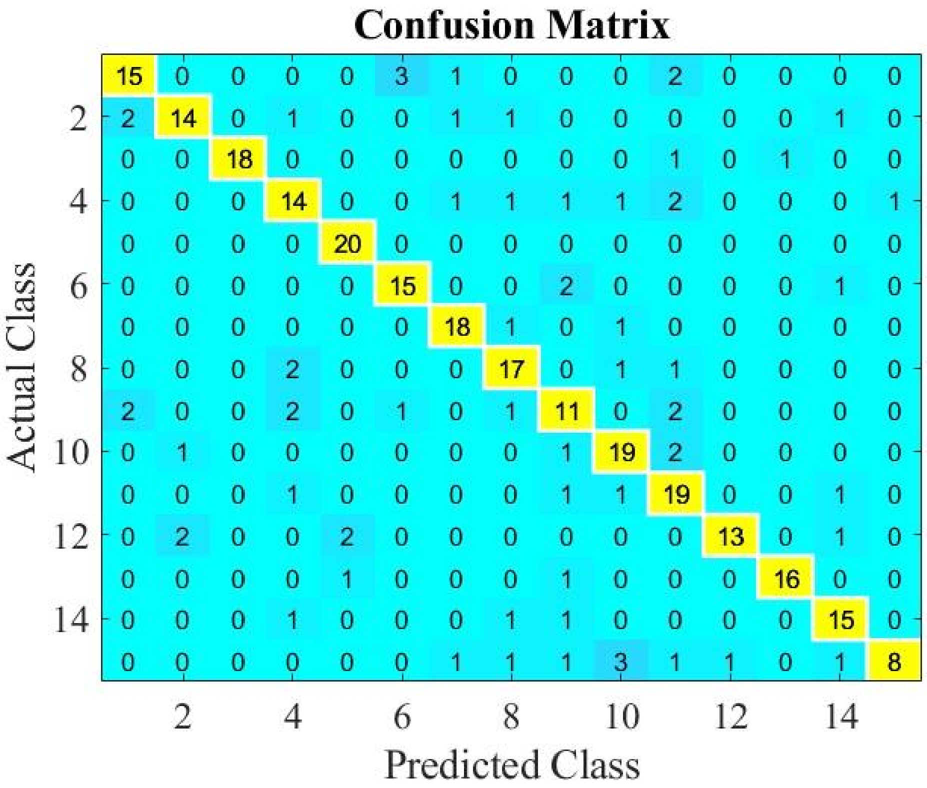
- Activation function = ‘relu’
- Hidden layer size = 2 layers with 10 neurons each
- Learning rate = $1.34*10^{-4}$.

For each above-mentioned model, a custom function was created to perform hyperparameter optimization. These functions returned a MATLAB structure containing the best model identified during the optimization process, the optimal hyperparameter configuration, and the highest accuracy achieved.

## 4.9 Testing methodology and metrics

In the following Section, the methodology employed to test the developed models, and the metrics used to measure their performance are presented.

For each model tested on the TRS, VS, or TS, a confusion matrix (CM) was generated. The CM is a cross table that records the number of occurrences between two raters: the true classification and the predicted classification. Each column of the matrix corresponds to instances of the actual classes, while each row represents the instances of the predicted classes, or vice-versa. The CM is crucial for deriving key performance metrics, such as accuracy, precision, recall, and F1-score, providing in-depth insights into the model's effectiveness across multiple classes. The classes are consistently arranged across both rows and columns, so correctly classified instances align along the main diagonal from the top left to the bottom right. These diagonal elements indicate where the true and predicted classifications match, indicating agreement between the two (Figure 4.9.1).



| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 0 | 0 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| 2 | 14 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 14 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 18 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 | 17 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 11 | 0 | 2 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 19 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 19 | 0 | 0 | 1 | 0 |
| 0 | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 16 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 15 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 3 | 1 | 1 | 0 | 1 | 8 |

*Figure 4.9.1: Illustration of a CM for multiclass classification problem, generated using the ECOC model with a RBF kernel. The total sum of all elements in the matrix equals the number of observations used in the classification task.*

The trained models were evaluated on unseen test data (TS), with the CM serving as the basis for calculating several useful evaluation metrics. The performance of the models was quantified using four key metrics: accuracy, F1-score, precision and recall [62]. In a multiclass setting, these metrics are often computed for each class individually, and then averaged to provide overall performance metrics. Two common averaging methods are:

- **Macro-Averaging**: Calculates the metrics for each class independently and then takes the average, giving equal weight to each class.
- **Weighted-Averaging**: Computes the metrics for each class and takes a weighted average, where the weights are based on the number of instances in each class.

In this thesis, the macro-averaging method was used.

Before describing these evaluation metrics, some definitions are necessary:

- **Correctly Classified** instances (CC): The diagonal elements of the CM, corresponding to the correct predictions for the i-th class.
- **Misclassified** instances (MC): The sum of the elements in the column of the i-th class, excluding the diagonal element, representing instances that were incorrectly classify as the i-th class.
- **False Positives** (FP): The sum of the elements in the rows of the i-th class, excluding the diagonal element, representing instances of the i-th class that were incorrectly predicted as other classes.

1. **Accuracy**: Accuracy is defined as the proportion of correctly predicted instances among the total instances. Specifically, it measures the ratio of the number of correctly classified instances to the total number of instances across all classes.

$$Accuracy = \frac{\sum_{i=1}^{N} CC_i}{\sum_{i=1}^{N}(CC_i + MC_i + FP_i)} \tag{33}$$

where N is the total number of classes, equal to 15.

2. **Precision**: Precision is the ratio of correctly predicted instances of the i-th class to the total instances predicted as the i-th class. It measures the accuracy of the positive predictions.

$$Precision_i\ (\%) = \frac{CC_i}{CC_i + FP_i} * 100 \tag{34}$$

3. **Recall**: Otherwise known as sensitivity, it is the ratio of correctly predicted instances of i-th class to the total instances that belong to the i-th class. It measures the model's ability to identify all relevant instances of a class.

$$Recall_i\ (\%) = \frac{CC_i}{CC_i + MC_i} * 100 \tag{35}$$

Since this is a multiclass classification task and all the classes must be considered, Macro Average Precision and Recall were computed as the arithmetic mean of the metrics for individual classes.

$$MacroAveragePrecision\ (\%) = \frac{\sum_{i=1}^{N} Precision_i}{N} \tag{36}$$

$$MacroAverageRecall\ (\%) = \frac{\sum_{i=1}^{N} Recall_i}{N} \tag{37}$$

where N is the total number of classes, equal to 15.

The F1-score with macro-averaging was used to evaluate the models' overall performance across all classes.

4. **F1-score**: The F1-score is the harmonic mean of Macro Average Precision and Recall, providing a single metric that balances the two.

$$F1score\ (\%) = 2 * \frac{(MacroAveragePrecision * MacroAverageRecall)}{(MacroAveragePrecision + MacroAverageRecall)} \tag{38}$$

These metrics have been computed for each CM, resulting in 12 sets of evaluation metrics being collected.

# 5 Results

In this chapter, the results of the testing stage are presented (see Section 4.9).

## 5.1 Classification performance

For each model, the optimal hyperparameters configuration (see Section 4.8) was selected based on the highest accuracy achieved on the VS. Additionally, TRS accuracy was assessed to determine if models' performance was influenced by overfitting. As described in Section 4.9, accuracy, recall, precision and F1-score were calculated to evaluate the performance for each model. The values for these evaluation metrics are presented in Table 5.1.1.

| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
|---|---|---|---|---|---|
| Decision Tree | TRS | 71,88 | 72,75 | 71,18 | 71,95 |
| | VS | 44,62 | 40,19 | 37,16 | 38,86 |
| | TS | 40,07 | 40,31 | 39,42 | 39,86 |
| Discriminant Analysis | TRS | 90,08 | 90,64 | 89,89 | 90,26 |
| | VS | 63,12 | 64,33 | 62,06 | 63,17 |
| | TS | 67,00 | 72,85 | 66,59 | 69,58 |
| ECOC with Ensemble template | TRS | 94,77 | 94,95 | 94,66 | 94,80 |
| | VS | 75,61 | 74,35 | 73,42 | 73,88 |
| | TS | 76,09 | 77,56 | 75,56 | 76,55 |
| ECOC with linear kernel | TRS | 89,68 | 88,25 | 89,53 | 89,65 |
| | VS | 67,44 | 68,76 | 69,28 | 69,02 |
| | TS | 70,37 | 70,62 | 70,39 | 70,50 |

| Model | Set | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|---|
| ECOC with polynomial kernel | TRS | 94,37 | 94,35 | 94,32 | 94,33 |
| | VS | 69,07 | 71,44 | 71,35 | 71,39 |
| | TS | 71,72 | 73,43 | 71,40 | 72,40 |
| ECOC with RBF kernel | TRS | 93,83 | 94,08 | 93,82 | 93,95 |
| | VS | 76,42 | 76,96 | 74,99 | 75,96 |
| | TS | 78,11 | 79,65 | 77,80 | 78,71 |
| Ensemble | TRS | 93,30 | 93,76 | 93,75 | 93,76 |
| | VS | 69,43 | 67,27 | 66,45 | 66,86 |
| | TS | 67,00 | 64,09 | 64,21 | 64,15 |
| k-NN | TRS | 81,92 | 83,58 | 81,64 | 82,60 |
| | VS | 67,56 | 71,88 | 69,47 | 70,66 |
| | TS | 72,39 | 74,81 | 72,04 | 73,40 |
| Naïve Bayes Classifier | TRS | 68,94 | 71,37 | 68,61 | 69,96 |
| | VS | 58,44 | 61,72 | 58,82 | 60,24 |
| | TS | 59,93 | 64,42 | 59,90 | 62,08 |
| Neural Network | TRS | 75,50 | 75,42 | 75,25 | 75,33 |
| | VS | 67,02 | 67,85 | 67,34 | 67,59 |
| | TS | 65,99 | 66,71 | 65,68 | 66,19 |

*Table 5.1.1:Values of the adopted classification metrics for the subjects' classification task obtained by each model on the complete dataset.*

For each model, the values of each metric over the three datasets are aggregated to derive the standard deviation values. These values are used to draw a bar diagram to allow a faster comparison between the performance of different models (Figure 5.1.1).

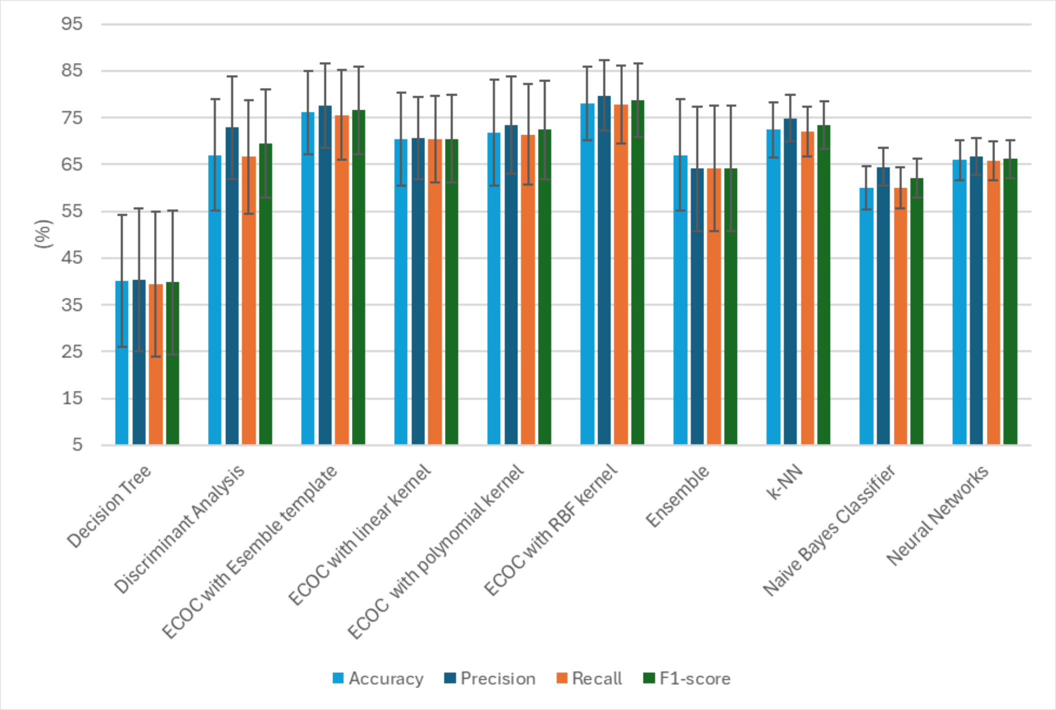


*Figure 5.1.1: Bar diagram that sums up the classification performance of the models on the complete dataset. The bars display the TS values ± standard deviation of the TRS, VS and TS. Each set of bars represents the corresponding metric values for each of the 10 models.*

To assess whether the models performed consistently across individual subjects, an analysis of the recognition percentages for each model on a per-subject basis was performed. This analysis revealed notable variations in recognition performance among subjects, indicating that the models did not perform identically across all individuals. Figure 5.1.2 illustrates the recognition performance of all models for each subject. To allow a better comprehension of the performance, Figure 5.1.3 shows the average recognition percentages of all models for each subject.

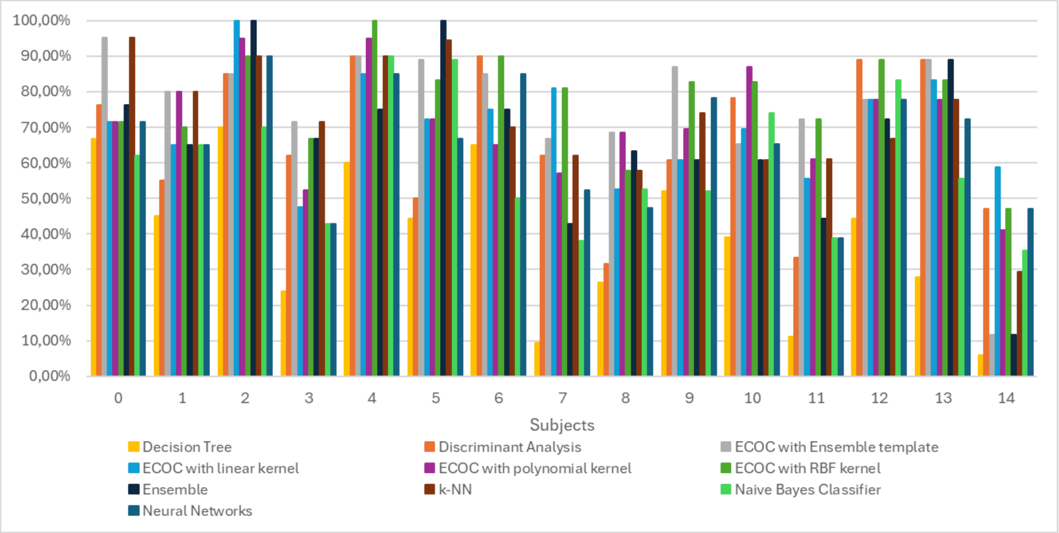


*Figure 5.1.2: Performance of each model across different subjects. Each group of bars denotes the values of the corresponding percentages for each of the ten classification models.*

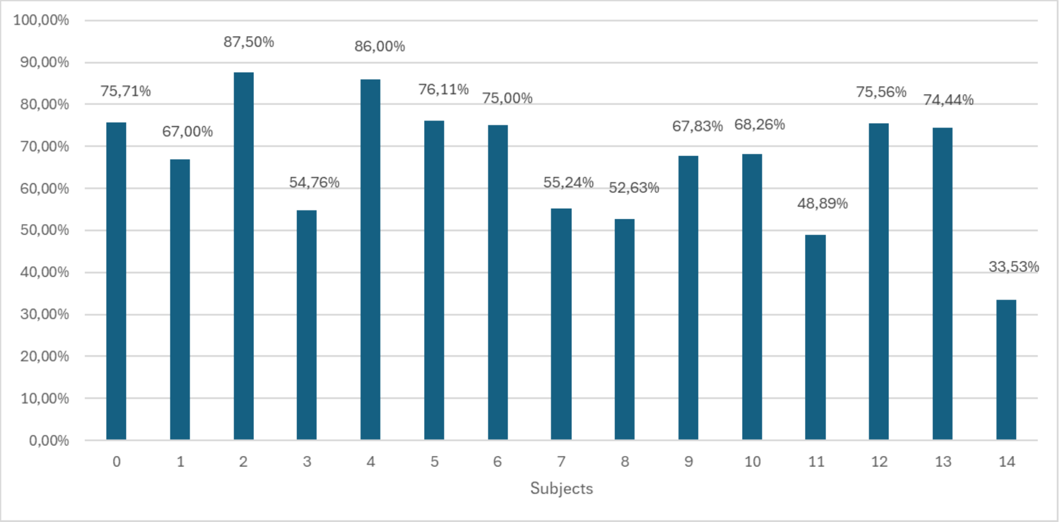


*Figure 5.1.3: Averaged recognition performance in percentage across different subjects. Each bar represents the average value between models for each subject.*

The study was also conducted on each of the three speeds to evaluate how the models performed depending on the speed range.

Table 5.1.2, Table 5.1.3, and Table 5.1.4 represent the values of accuracy, precision, recall, and F1-score obtained for normal speed, slow speed, and fast speed datasets respectively.

| **Normal speed** | | | | | |
|---|---|---|---|---|---|
| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| Decision Tree | TRS | 76,58 | 80,27 | 76,63 | 78,40 |
| | VS | 41,04 | 24,67 | 23,89 | 24,27 |
| | TS | 31,25 | 33,24 | 31,07 | 32,12 |
| Discriminant Analysis | TRS | 85,27 | 87,16 | 85,31 | 86,23 |
| | VS | 73,95 | 73,97 | 73,89 | 73,93 |
| | TS | 72,92 | 76,75 | 72,71 | 74,68 |
| ECOC with Ensemble template | TRS | 56,49 | 59,88 | 55,84 | 57,78 |
| | VS | 75,07 | 66,11 | 68,33 | 67,20 |
| | TS | 73,96 | 76,83 | 74,48 | 75,64 |
| ECOC with linear kernel | TRS | 85,26 | 86,91 | 84,76 | 85,82 |
| | VS | 69,07 | 57,41 | 58,33 | 57,87 |
| | TS | 76,04 | 77,77 | 76,64 | 77,20 |
| ECOC with polynomial kernel | TRS | 94,79 | 95,63 | 94,6 | 95,11 |
| | VS | 68,29 | 75,78 | 73,89 | 74,82 |
| | TS | 75,00 | 80,95 | 75,71 | 78,24 |

| Normal speed | | | | | |
|---|---|---|---|---|---|
| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| ECOC with RBF kernel | TRS | 86,13 | 88,10 | 85,78 | 86,92 |
| | VS | 72,92 | 55,44 | 58,33 | 56,85 |
| | TS | 86,46 | 89,55 | 86,44 | 87,97 |
| Ensemble | TRS | 93,05 | 93,34 | 92,62 | 93,47 |
| | VS | 64,72 | 55,39 | 60,00 | 57,60 |
| | TS | 65,62 | 68,09 | 67,32 | 67,70 |
| k-NN | TRS | 73,59 | 77,18 | 73,22 | 75,15 |
| | VS | 71,72 | 61,78 | 67,78 | 64,64 |
| | TS | 73,96 | 78,30 | 75,03 | 76,63 |
| Naïve Bayes Classifier | TRS | 91,33 | 92,02 | 92,14 | 91,61 |
| | VS | 72,61 | 72,89 | 71,11 | 71,99 |
| | TS | 59,38 | 60,93 | 58,92 | 59,91 |
| Neural Network | TRS | 87,43 | 89,23 | 87,17 | 88,18 |
| | VS | 70,82 | 63,33 | 67,22 | 65,22 |
| | TS | 73,96 | 73,08 | 73,67 | 73,37 |

*Table 5.1.2: Values of the adopted classification metrics for the subjects' classification task obtained by each model considering only normal speed range.*

For each model, the values of each metric over the three datasets are aggregated to calculate the standard deviation values. These values are used to draw a bar diagram to allow a faster comparison between the performance of different models (Figure 5.1.4).

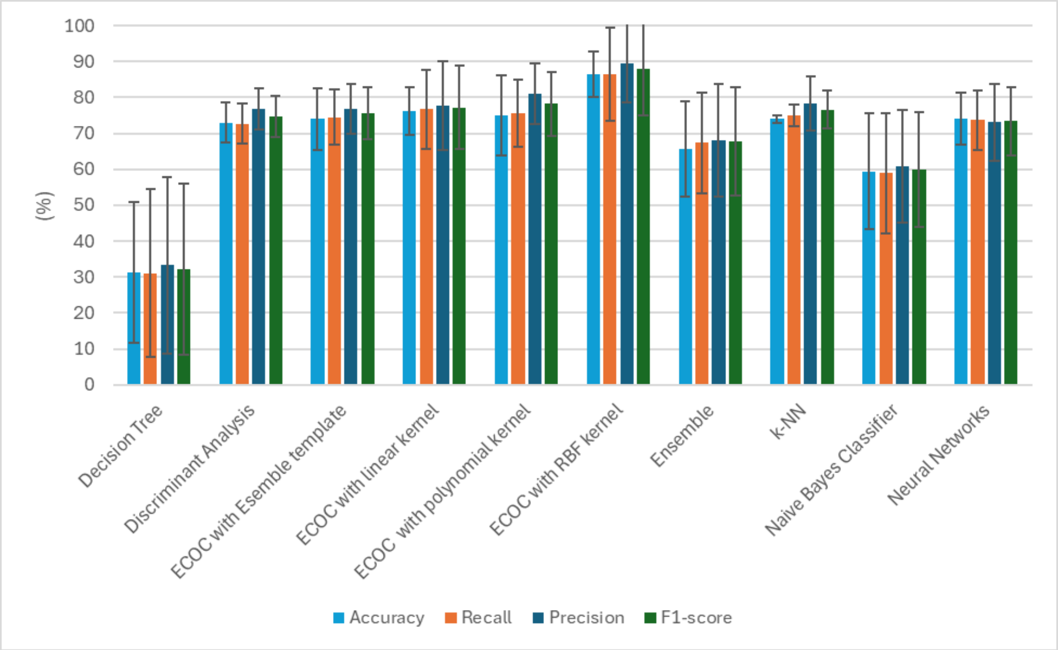


*Figure 5.1.4: Bar diagram that sums up the classification performance of the models for the normal speed range. The bars display the TS values ± standard deviation of the TRS, VS and TS. Each set of bars represents the corresponding metric values for each of the 10 models.*

Similarly to the analysis conducted on the complete dataset, to assess whether the models performed consistently across individual subjects, the recognition percentages for each model were evaluated on a per-subject basis. This analysis revealed notable variations in recognition performance among subjects, indicating that the models did not perform identically across all individuals. Figure 5.1.5 illustrates the recognition performance of all models for each subject. For a faster comprehension, Figure 5.1.6 shows the average recognition percentage of all models for each subject.

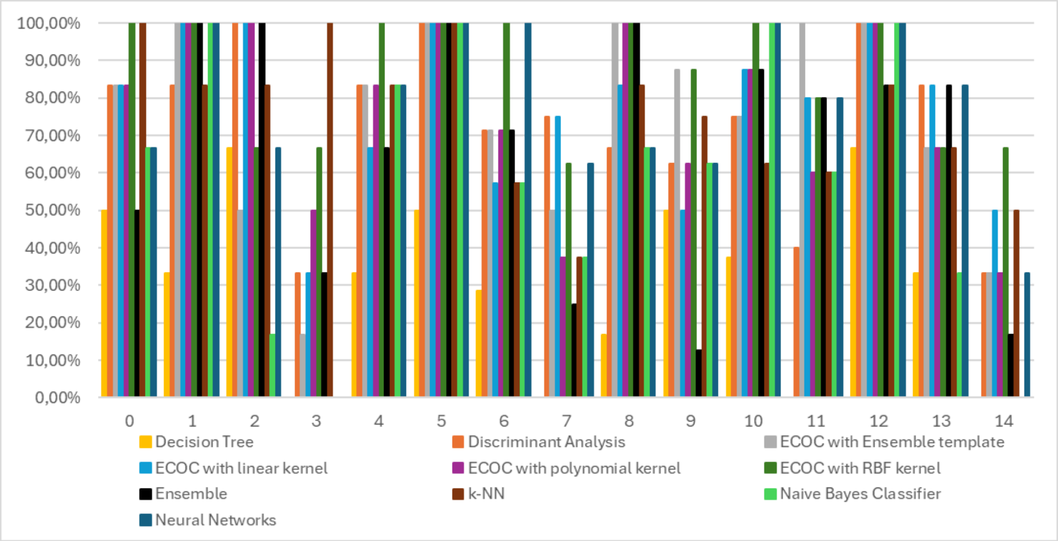


*Figure 5.1.5: Performance of each model across different subjects for the normal speed range. Each group of bars denotes the values of the corresponding percentages for each of the ten classification models.*

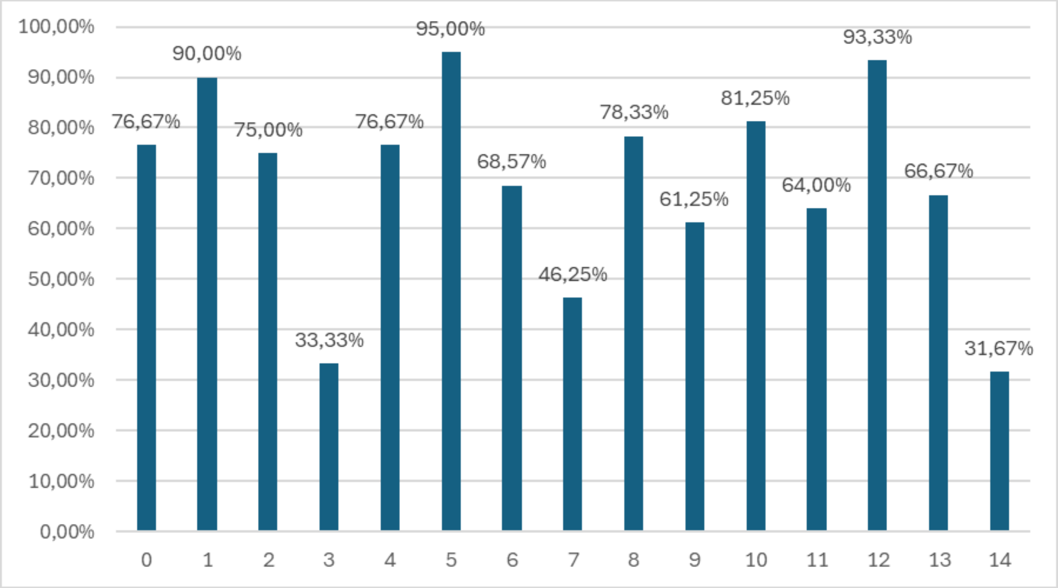


*Figure 5.1.6: Averaged recognition performance in percentage across different subjects for the normal speed range. Each bar represents the average value between models for each subject.*

| Slow speed | | | | | |
|---|---|---|---|---|---|
| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| Decision Tree | TRS | 73,30 | 81,47 | 78,6 | 80,01 |
| | VS | 47,64 | 61,16 | 54,11 | 57,42 |
| | TS | 43,11 | 42,28 | 42,6 | 42,44 |
| Discriminant Analysis | TRS | 82,79 | 85,1 | 82,53 | 83,79 |
| | VS | 74,24 | 82,03 | 78,89 | 80,43 |
| | TS | 64,12 | 73,49 | 64,17 | 68,51 |
| ECOC with Ensemble template | TRS | 59,84 | 67,32 | 59,98 | 63,44 |
| | VS | 72,48 | 74,06 | 69,22 | 71,56 |
| | TS | 76,34 | 74,52 | 74,81 | 74,66 |
| ECOC with linear kernel | TRS | 89,91 | 90,66 | 89,48 | 90,07 |
| | VS | 71,59 | 82,13 | 75,78 | 78,83 |
| | TS | 71,76 | 74,4 | 72,07 | 73,22 |
| ECOC with polynomial kernel | TRS | 93,48 | 93,85 | 93,55 | 93,7 |
| | VS | 65,69 | 74,06 | 67,22 | 70,48 |
| | TS | 69,47 | 72,31 | 68,47 | 70,34 |
| ECOC with RBF kernel | TRS | 94,07 | 94,77 | 94,14 | 94,46 |
| | VS | 74,53 | 83,65 | 79,44 | 81,49 |
| | TS | 71,76 | 74,08 | 71,45 | 72,74 |

| Slow speed | | | | | |
|---|---|---|---|---|---|
| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| Ensemble | TRS | 92,29 | 92,69 | 92,2 | 92,44 |
| | VS | 62,10 | 62,81 | 60,22 | 61,49 |
| | TS | 69,47 | 73,41 | 68,82 | 71,04 |
| k-NN | TRS | 73,89 | 79,15 | 72,44 | 75,62 |
| | VS | 69,48 | 75,58 | 74,22 | 74,89 |
| | TS | 67,94 | 74,48 | 66,8 | 70,43 |
| Naïve Bayes Classifier | TRS | 85,46 | 87,26 | 85,51 | 86,38 |
| | VS | 62,68 | 66,38 | 61,78 | 63,99 |
| | TS | 65,65 | 74,33 | 65,18 | 69,46 |
| Neural Network | TRS | 93,77 | 94,27 | 93,67 | 93,97 |
| | VS | 72,42 | 76,02 | 71,89 | 73,89 |
| | TS | 64,12 | 67,96 | 63,81 | 65,82 |

*Table 5.1.3: Values of the adopted classification metrics for the subjects' classification task obtained by each model considering only slow speed range.*

Figure 5.1.7 displays the bar chart with the standard deviation values obtained from aggregating each evaluation metric over the three datasets for each model.

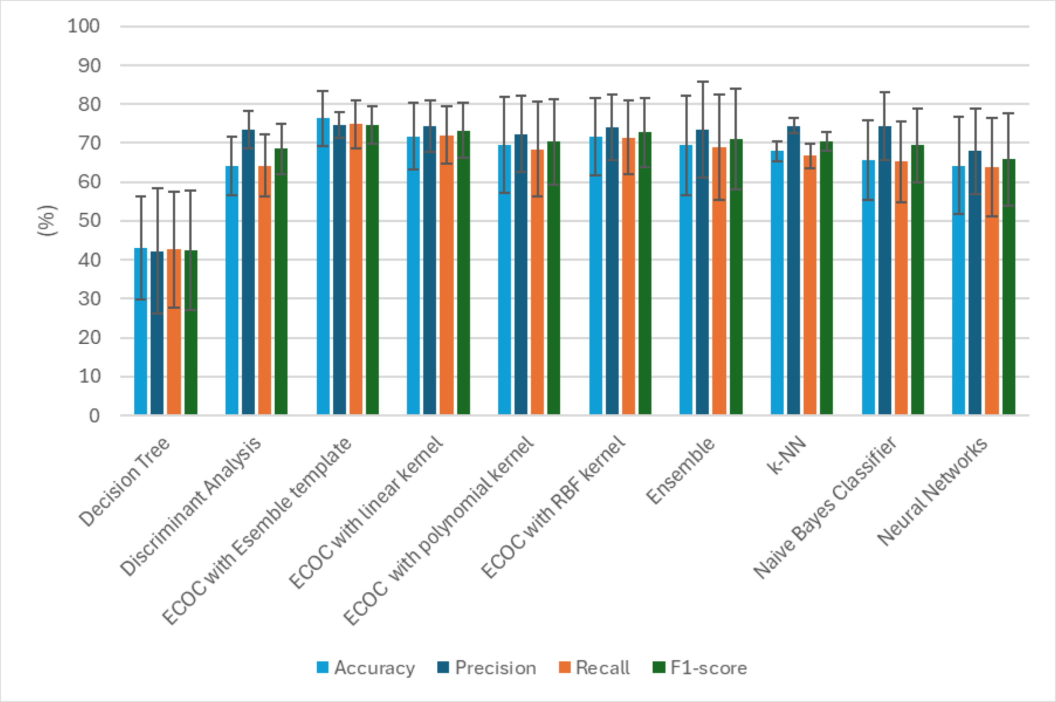


*Figure 5.1.7: Bar diagram that sums up the classification performance of the models for the slow speed range. The bars display the TS values ± standard deviation of the TRS, VS and TS. Each set of bars represents the corresponding metric values for each of the 10 models.*

In this instance as well, the recognition percentages for each model were evaluated on a per-subject basis. This analysis revealed notable variations in recognition performance among subjects, indicating that the models did not perform identically across all individuals. Figure 5.1.8 illustrates the recognition performance of all models for each subject. For a faster comprehension, Figure 5.1.9 shows the average recognition percentage of all models for each subject.

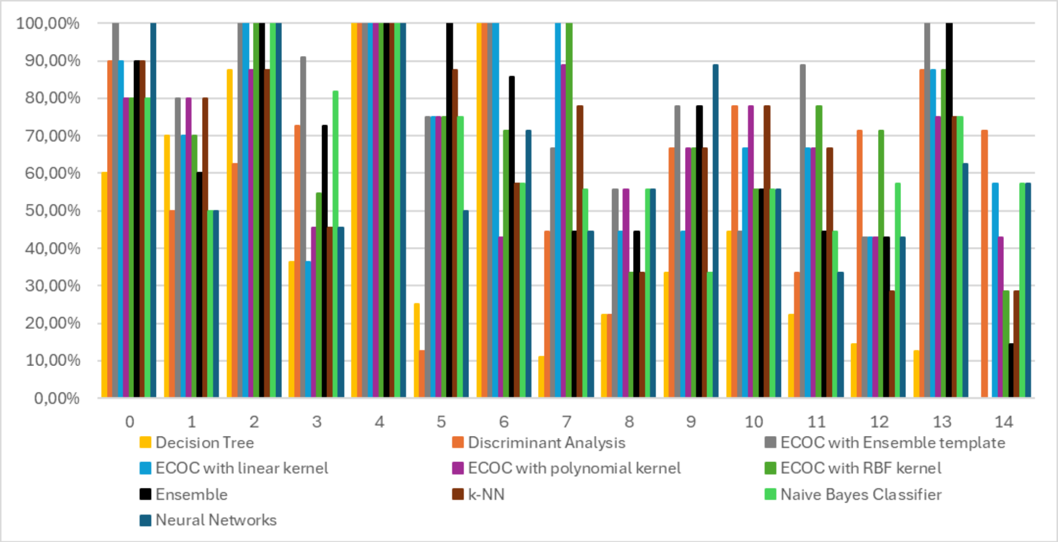


*Figure 5.1.8: Performance of each model across different subjects for the slow speed range. Each group of bars denotes the values of the corresponding percentages for each of the ten classification models.*

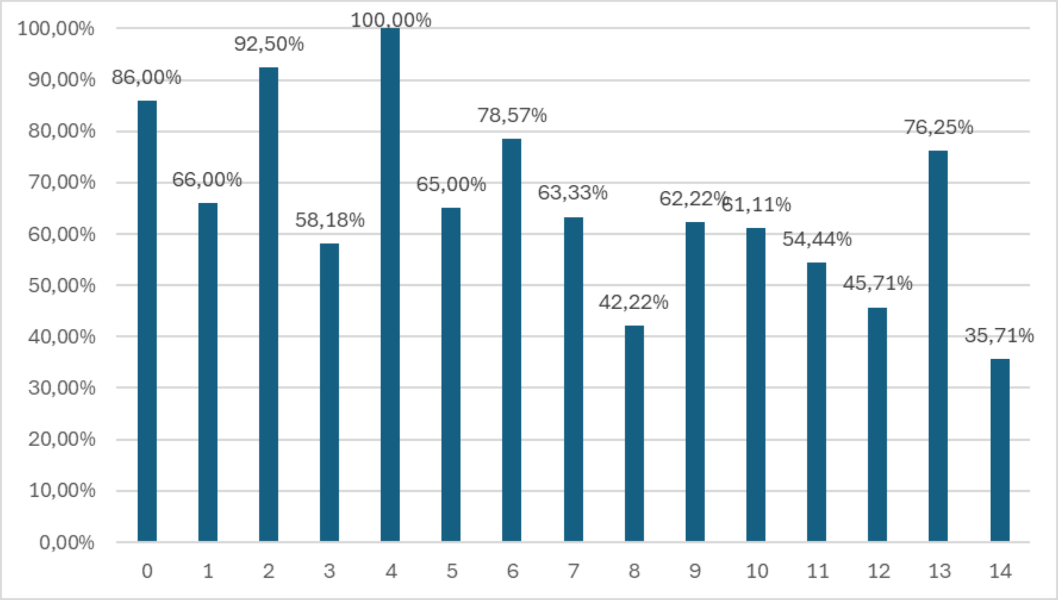


*Figure 5.1.9: Averaged recognition performance in percentage across different subjects for the slow speed range. Each bar represents the average value between models for each subject.*

| Fast speed | | | | | |
|---|---|---|---|---|---|
| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| Decision Tree | TRS | 77,64 | 78,69 | 76,96 | 77,81 |
| | VS | 57,97 | 54,22 | 61,11 | 57,46 |
| | TS | 45,71 | 46,93 | 42,89 | 44,82 |
| Discriminant Analysis | TRS | 89,93 | 90,28 | 89,45 | 89,86 |
| | VS | 74,49 | 77,78 | 77,78 | 77,78 |
| | TS | 64,29 | 67,11 | 61,56 | 64,21 |
| ECOC with Ensemble template | TRS | 46,18 | 56,94 | 45,29 | 50,45 |
| | VS | 75,74 | 83,89 | 83,33 | 83,61 |
| | TS | 78,57 | 74,02 | 76,11 | 75,05 |
| ECOC with linear kernel | TRS | 89,89 | 90,12 | 89,69 | 89,9 |
| | VS | 66,20 | 44,44 | 50,00 | 47,06 |
| | TS | 60,00 | 53,24 | 57,44 | 55,26 |
| ECOC with polynomial kernel | TRS | 94,38 | 94,79 | 94,27 | 94,53 |
| | VS | 69,49 | 67,22 | 67,78 | 67,50 |
| | TS | 71,43 | 74,13 | 68,78 | 71,35 |
| ECOC with RBF kernel | TRS | 94,38 | 94,54 | 94,38 | 94,46 |
| | VS | 73,36 | 65,00 | 67,78 | 66,36 |
| | TS | 78,57 | 81,03 | 76,33 | 78,61 |

| **Fast speed** | | | | | |
|---|---|---|---|---|---|
| **Model** | **Set** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| Ensemble | TRS | 93,63 | 97,26 | 96,47 | 96,86 |
| | VS | 78,53 | 82,33 | 80,00 | 81,15 |
| | TS | 64,29 | 59,92 | 62,00 | 60,94 |
| k-NN | TRS | 95,50 | 95,35 | 95,18 | 95,27 |
| | VS | 77,02 | 75,56 | 74,44 | 75,00 |
| | TS | 78,57 | 76,10 | 77,78 | 76,93 |
| Naïve Bayes Classifier | TRS | 87,66 | 88,42 | 87,03 | 87,72 |
| | VS | 57,84 | 39,89 | 47,78 | 43,48 |
| | TS | 50,00 | 51,26 | 48,00 | 49,58 |
| Neural Network | TRS | 92,69 | 93,11 | 92,31 | 92,71 |
| | VS | 67,99 | 56,67 | 58,89 | 57,76 |
| | TS | 58,57 | 61,77 | 56,33 | 58,93 |

*Table 5.1.4: Values of the adopted classification metrics for the subjects' classification task obtained by each model considering only fast speed range.*

Figure 5.1.10 illustrates the bar charts with the standard deviation values resulting from the aggregation of each evaluation metric over the three datasets for each model.

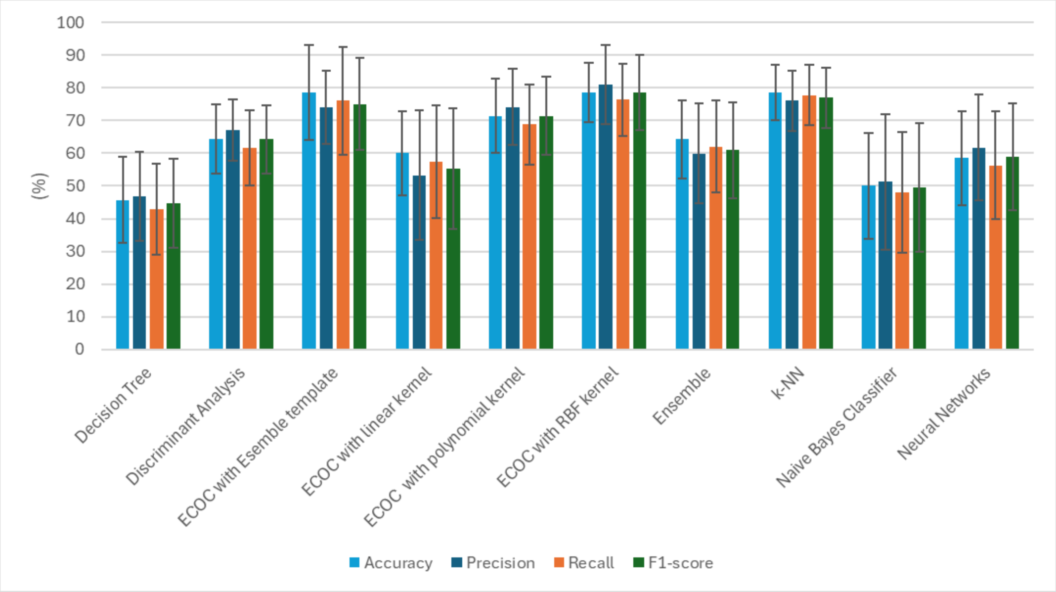


*Figure 5.1.10: Bar diagram that sums up the classification performance of the models for the fast speed range. The bars display the TS values ± standard deviation of the TRS, VS and TS. Each set of bars represents the corresponding metric values for each of the 10 models.*

To assess whether the models performed consistently across individual subjects, the recognition percentages for each model were evaluated on a per-subject basis. This analysis revealed notable variations in recognition performance among subjects, indicating that the models did not perform identically across all individuals. Figure 5.1.11 illustrates the recognition performance of all models for each subject. To allow a better comprehension, Figure 5.1.12 shows the averaged recognition percentages for each model over the fifteen subjects.

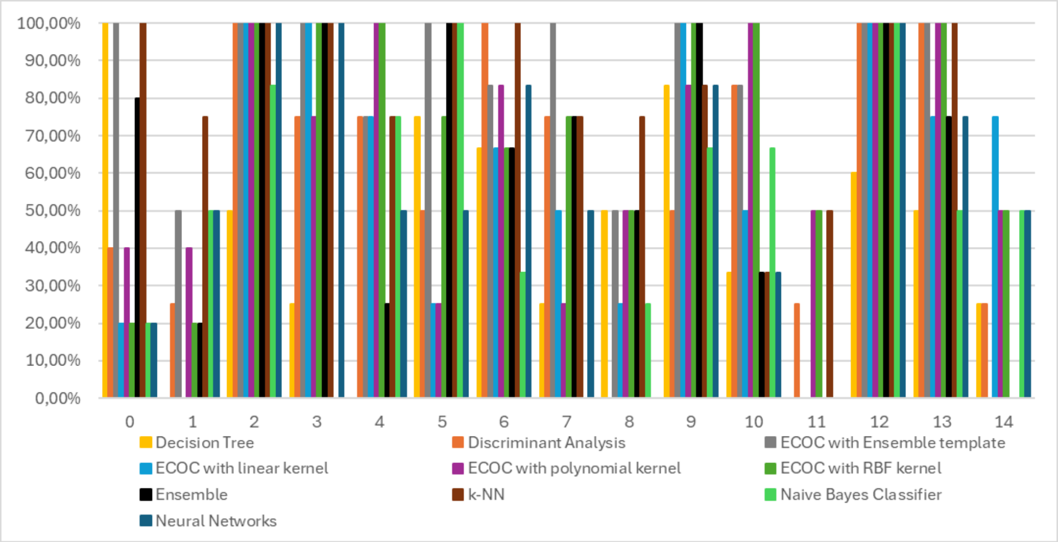


*Figure 5.1.11: Performance of each model across different subjects for the fast speed range. Each group of bars denotes the values of the corresponding percentages for each of the ten classification models.*

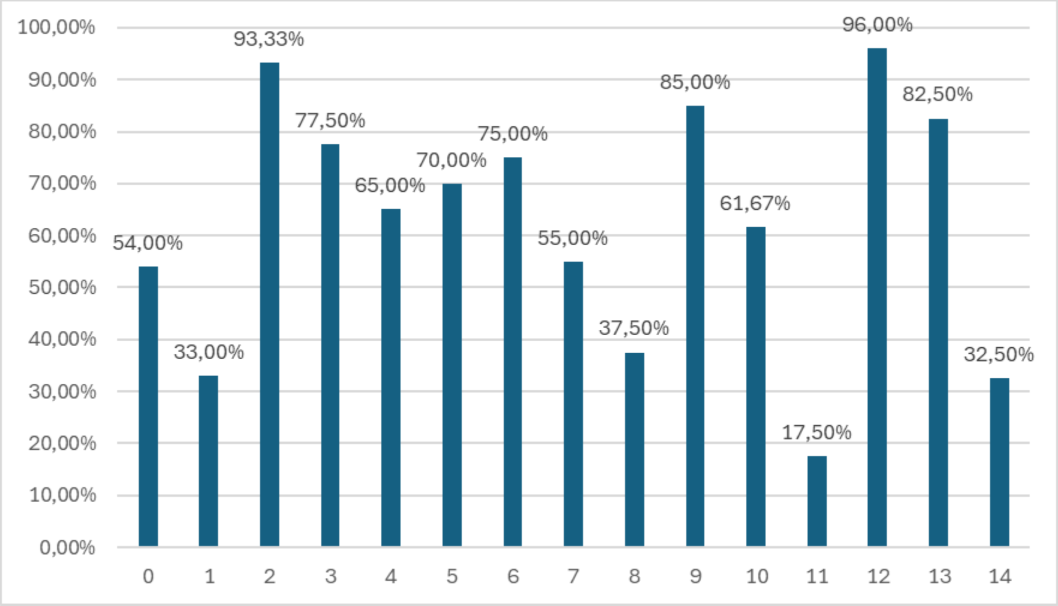


*Figure 5.1.12: Averaged recognition performance in percentage across different subjects for the fast speed range. Each bar represents the average value between models for each subject.*

# **6** Discussions

This work aimed at developing a model-based method using biomechanical domain-specific features through a multi-camera marker-less system for automatic gait recognition. Data were collected from fifteen young healthy subjects walking at three different speed ranges on both the sagittal and frontal planes, using two Azure Kinect sensors for right and left-side views, and one Vicon VUE camera for frontal views. The acquired data were processed to extract biomechanical features which were then used to train ten ML models with different TRS/VS combinations. Before training the models, the most relevant gait features were selected using the automatically using the mRMR. Initially the models were tested on the entire dataset, covering data across all three speed ranges. Then, they were evaluated separately on each speed range to assess model performance specific to each speed.

## 6.1 Selected features and models' feature subsets

Table 4.7.2.1 reports the nineteen biomechanical features that were selected through the mRMR algorithm. The most relevant features included kinematics quantities, frequency parameters, and correlation indices. The features were categorized as follows: 2 associated with the head, 2 with the lower limbs, and 15 with the upper limbs, highlighting the significant role of the upper limbs in gait recognition tasks. Among these features, 10 belonged to the time domain and 9 to the frequency domain, demonstrating a balanced distribution between the two domains. Furthermore, there was a predominance of features related to the sagittal plane, with 12 out of 19 features belonging within this plane. It could also be observed that the extracted symmetry indices were not particularly significant, as only ASA of the swing appeared among the 19 features. This was likely because the subjects involved in the project are healthy; therefore, there were no substantial differences between one subject and another in terms of symmetry.

For most models, the feature subset that led to the highest accuracy through wrapper feature selection was the first one, which included all 19 features together. In contrast, the Ensemble classifier achieved the highest accuracy with the second feature subset, which considered the first 18 features (Table 4.7.2.2).

## 6.2 Models' performance

Classification metrics provided a detailed measure of each model's performance for complete dataset (Table 5.1.1) and each speed range (normal, Table 5.1.2; slow, Table 5.1.3; fast, Table 5.1.4).

For all models trained on the complete dataset, accuracy was higher on the TRS compared to the values achieved when tested on unseen data of the TS, as expected. This difference highlights that the models successfully learned patterns from the training data, but they struggled to generalize effectively to unseen data, suggesting some degree of overfitting, especially in those models where the accuracy gap between TRS and TS data was significant. Additionally, precision and recall attained approximately the same percentages within the same model for both TRS and TS data. This consistency demonstrates that the models were able to manage the trade-off between precision (the ability to assign strides to the correct subject) and recall (the ability to identify all strides that belong to the same subject). Especially for ECOC models, precision and recall were closely aligned across TRS and TS, meaning that the models were not only accurate in identifying relevant instances but also effective at ensuring that all relevant instances are captured, even when faced with unseen data. ECOC with RBF kernel obtained the highest values on the TS (Figure 5.1.1), 78,1% accuracy and 78,7% F1-score, making it the top-performing model. Decision Tree achieved an accuracy value of 40% and a F1-score of 39,86%, making it the worst-performing model overall. The poor performance of Decision Tree can be attributed to its tendency to overfit the training data, its sensitivity to small changes, and its reliance on single-feature splits, that limits its ability to capture complex gait patterns compared to ensemble methods such as ECOC that combine multiple features simultaneously.

For models trained on the normal, slow, and fast speed datasets, performances were similar to the performances on the complete dataset. Accuracy was higher on the TRS compared to the values achieved when tested on unseen data, except for the ECOC with Ensemble template that obtained lower accuracy values on the TRS with respect to the values obtained on the TS, meaning that the model struggled during training phase, likely indicating a low capacity for capturing patterns in the training data.  ECOC with RBF kernel achieved the best values for accuracy and F1-score on the TS both for normal (86,46% and 87,97%, respectively) and fast speed ranges (78,57% and 78,61%, respectively); ECOC with Ensemble template reached the best values for accuracy and F1-score for slow speed range, 76,34% and 74,66% respectively. This suggests that while ECOC model is the most robust across all walking speeds, it benefits from the natural, consistent gait patterns observed at normal speed, whereas the variations and irregularities in slow and fast walking speeds reduce its performance. Across all the three walking speeds, Decision Tree was the worst model with accuracy and F1-score values always under 45% on the TS, making it inappropriate for this type of task.

In general, the models that performed better across all the three datasets also demonstrated a balance between precision and recall, demonstrating their ability to correctly classify subjects and capture relevant strides efficiently.

Bar chart in Figure 5.1.4 highlighted that ECOC with RBF kernel outperformed all other models in the normal speed range. For most models recall rates were lower compared to accuracy and precision. Figure 5.1.7 showed that all models, but Decision Tree, had performance around 70% in the slow speed range. Most models showed higher precision compared to recall, indicating that while they could accurately identify the positive class, they might miss some instances, particularly Naïve Bayes, which showed a precision spike but lower recall. Figure 5.1.10 displayed models' performance on the TS of the fast speed range, and it can be observed that these performances were less balanced across models with respect to the normal and slow speed datasets. ECOC with RBF kernel was the best performer, followed by k-NN with very near values.

A thorough understanding of how each model processed subject-specific gait data was provided by the TS confusion matrix analysis of model performance across individual subjects. These results were represented through bar charts, one for each dataset. Each chart has a group of ten bars for each subject that denotes the values of the corresponding percentages for each of the ten classification models.

For what concerns the complete dataset (Figure 5.1.2), the ECOC models, especially the one with RBF kernel (green bar) and Ensemble template (grey bar), consistently outperformed other models across most subjects. These models often reached performance around 90% on many subjects, demonstrating strong generalization across various subjects. The best performance was observed for subject 2 and subject 4, where almost all models performed well. For subject 2 the Ensemble method (blue bar) and ECOC with linear kernel (light blue bar) reached 100% of recognition rate; for subject 4 the ECOC with the RBF kernel reached 100% of recognition rate; for subject 5 the Ensemble method achieved 100%

recognition rate. However, for some subjects like 8, 11, and 14 performances dropped below 60% for most models, particularly subject 14 who was the most difficult to classify correctly for all models, included the top-performing ECOC models. Some subjects like 1, 5, 12, and 13 exhibited high variability in performance across models. While top performers like ECOC still performed well, other models like Decision Tree (yellow bar) struggled, with performance dropping significantly. For subjects 3, 7 and 11, most models struggled to reach 70%. In particular, Decision Tree never reached performance over 70% across all subjects, further confirming its unsuitability for this type of classification task. Figure 5.1.3 represents the average recognition percentage of each model for each subject and helps in understanding that the models' performance is deeply related to the fact that some subjects are less recognizable than others.

The possible causes that led some models to achieve very low performance on specific subjects can be the presence of outliers, the homogeneity of gait characteristics among subjects, and the poor generalization ability of the models. Specifically, subject 14 represents a case where data variability was not well managed by the models. Simpler models, such as Decision Tree and Naïve Bayes, tend to suffer more from the presence of outliers, while more robust models, like ECOC and Ensemble methods, can handle such anomalies better due to their more complex structure, leading to better performance even in these cases.

Another factor that could explain low performance on some subjects is the excessive homogeneity of gait characteristics among subjects in the dataset. If the data shows little variability between subjects, the models struggle to distinguish individuals correctly. For example, subjects like 3, 7, and 11 have the extracted gait characteristics too similar to those of other subjects, making it difficult for the models to recognize significant differences. This problem is pronounced in simpler models like Decision Tree, but even more advanced models like Discriminant Analysis and k-NN showed significant performance drops in these cases, suggesting that the homogeneity of the data made it difficult for these algorithms to correctly separate subjects based on the extracted biometric features. For example, subject 3 was classified as subject 7 for 18 times, and as subject 10 for 16 times, while subject 7 was classified as subject 3 for 37 times in total across all models. Additionally, subject 11 was classified as subject 4 for 24 times, and as subject 3 for 16 times in total across all models. This means that the models were not able to correctly distinguish between them.

The poor generalization ability is particularly evident for models like the Ensemble, which performed well on the training set but experienced a drastic drop when applied to unseen data. This drop can be linked to the model's complexity, making it too specialized on training data and unable to generalize well on subjects with slightly different or more variable gait patterns. Models like Discriminant Analysis and k-NN also showed generalization difficulties for some subjects, especially with subjects 8, 11, and 14 where performance dropped significantly below 60%.

Concerning the normal range of speed (Figure 5.1.5), models performed much better than the complete dataset on each subject as confirmed by the metric values obtained on this range of speed. ECOC models were the top performing in most subjects, followed by Discriminant Analysis and k-NN. Also in this case Decision Tree was the worst model, as it never correctly classified four out of fifteen subjects. Many models reached 100% recognition percentages for different subjects, for example, ECOC with linear kernel and Ensemble method perfectly classified 4 subjects; ECOC with RBF kernel achieved 100% recognition percentage for 8 subjects; ECOC with polynomial kernel and ECOC with Ensemble template perfectly classified 5 subjects; Discriminant Analysis and k-NN perfectly classified 3 subjects. Subjects 3, 7, 11 and 14 were never classified in their class for at least one model, in particular subjects 3 had a recognition percentage of 0% in three models (Decision Tree, Naïve Bayes Classifier, and Neural Networks), while subject 14 in two models (Decision Tree, and Naïve Bayes Classifier). These observations were confirmed by the bar chart in Figure 5.1.6, in which subjects 3, 7, and 14 had an average recognition rate under 50%, while all other subjects were over 60%.

Regarding the slow speed range (Figure 5.1.8), there was much more variability for each model across subjects. Also in this case, ECOC models were the ones that obtained the highest values in percentage of recognition among subjects except for subject 14, while Decision Tree was the worst model. Subject 4 obtained 100% of recognition percentage in each model, meaning that the models always distinguished his gait characteristics among the ones of other subjects on the slow speed. Additionally, subject 2 and subject 6 that were perfectly classified by six and four models, respectively. On average, models on this range of speed performed worse than the normal range of speed. As highlighted in Figure 5.1.9, only three out of fifteen subjects reached a recognition rate over 85%. Subjects 6 and 13 reached a recognition rate of 78,57% and 76,25%, respectively, while all the remaining subjects achieved a recognition rate under 66%, meaning that for most subjects, gait characteristics at slow speed were not easily generalizable across individuals.

Referring to the fast speed range (Figure 5.1.11), there were less balance across models with respect to normal and slow speed datasets. Seven subjects were never classified correctly at least one time. For example, subject 11 was never classified in his class for 6 times, meaning that his gait characteristics were not easily recognized by the models. The model that achieved the highest recognition rate for most subject was the k-NN, followed by ECOC with Ensemble template and ECOC with RBF kernel. Averaged recognition performance in percentage across subjects (Figure 5.1.12) showed that only a few subjects were correctly recognized from the models, since only subjects 2 and 12 reached a percentage over 90%. Other five subjects (3, 5, 6, 9, and 13) reached a percentage over 70%. The remaining eight subjects achieved a percentage under 70%. Subjects 1, 8, and 14 reached percentages around 35%.

Differences in performance across speed datasets, in particular for slow and fast speeds, were certainly due to the presence of outliers and the poor generalization ability of the models, but at such unusual walking speeds for most of the subjects, their gait characteristics became similar and denatured, making it difficult for the models to generalize for each individual. In addition, depending on how slow or fast a subject walked in the tests, each subject was much more or much less represented in the datasets, respectively. This means that the models might specialized more on certain subjects with respect to others.

# 7 Conclusions

## 7.1 Main findings

The present study aimed to explore the potential for recognizing individuals through their unique gait patterns in an indoor environment. To this aim, a model-based method driven by biomechanical features using a multi-camera marker-less system for automatic gait recognition is developed. To reduce the computational cost of the model-based approach, features were selected automatically from a wide range of parameters from the literature, covering various domains and modalities.

The study focuses on identifying the most relevant biomechanical features from marker-less video data to recognize people by their gait using different machine learning models.

Fifteen healthy young subjects were involved in the study, walking at three self-selected speed ranges on a 5-meter walkway in an indoor environment wearing a T-shirt and shorts. Three RGB cameras, two Azure Kinect sensors and one Vicon VUE camera, were positioned laterally and frontally to the walkway to acquire data on the sagittal and frontal planes.

The proposed pipeline is based on seven steps (see Section 4), starting from the semi-automatical joint centres tracking with MoveNet, a deep-learning pose estimator. Other steps concern automatic stride segmentation to allow a stride-by-stride extraction of time-domain and frequency-domain features, outlier removals, features ranking with the mRMR algorithm, top-down wrapper feature selection with seven classification models, and models' hyperparameters tuning by grid-search and 5-folds cross-validation.

The results show that the most relevant features derived from mRMR algorithm were nineteen, including kinematic quantities, frequency parameters, and correlation indices. Fifteen out of the nineteen features belonged to the upper limbs with a balance between time- and frequency-domain. This finding is relevant because, in security applications, lower limbs can be easily hidden in crowds, whereas in clinical applications, such as Parkinson's disease, studies have demonstrated that the disease progression is often followed by a loss of upper limbs coordination [57]. Selected features have demonstrated a potential in discriminating individuals from their gait, especially when subjects walked at a self-selected normal pace, meaning that if models were trained on more data of the same walking speed, they could potentially perform better.

The model that demonstrated the best performance with the selected features was ECOC with RBF kernel for the overall dataset, and for both normal and fast speed datasets. The model that best performed for slow speed dataset was the ECOC with Ensemble template, meaning that the ECOC model is the most robust among the examined ML models for gait recognition tasks especially at normal walking speed where natural gait patterns were more consistent. ECOC models also demonstrated the precision-recall balance that is essential for reliable performance. Choosing robust models that can generalize well across different subjects and conditions is essential since some models' performance was affected by poor generalization capabilities for certain subjects and overfitting.

In conclusion, the identified feature subsets provide a strong basis for discriminating individuals from their gait with the ECOC model proving to be the most robust approach across different walking speeds. On the other hand, challenges such as overfitting, subject-specific performance variability, and decreased accuracy at both slow and fast speeds highlight the need for more extensive training data and further models’ optimization. Overall, these findings provide a strong foundation for future improvements in gait recognition systems.

## 7.2 Future works and related research

In the future, the implementation of larger and more diverse datasets among subjects should be addressed. In addition, further research could explore advanced feature selection methods, considering more complex interactions between features. Concerning security applications, the method could be improved introducing gait covariates in the dataset, such as carrying conditions or clothing variations, and real-time conditions.

This approach can be used in future studies to detect early-stage symptoms or subtle alterations in gait patterns that may signal disease onset, progression, or treatment response. Following surgery, an injury, or a stroke, patients' recovery can be tracked with the use of gait recognition devices. Clinicians can more accurately assess the success of rehabilitation programs and make necessary therapy adjustments by monitoring patients' gaits over time. It may be possible to create automated devices that offer real-time feedback on irregularities in gait, which would help patients and medical professionals optimize therapy.

# Bibliography

[1] A. Ross e A. K. Jain, «Human recognition using biometrics: An overview.,» *Annals of Telecommunications,* vol. 62, n. 1, pp. 11-35, 2007.

[2] P. Connor e A. Ross, «Biometric recognition by gait: A survey of modalities and features,» *Computer Vision and Image Understanding,* vol. 167, pp. 1-27, 2018.

[3] C. Wan, L. Wang e V. V. Phoha, «A Survey on Gait Recognition,» *ACM Computing Surveys,* vol. 51, n. 5, pp. 1-35, 2018.

[4] D. Hodgins, «The importance of measuring human gait,» *Med Device Technol.,* vol. 19, n. 5, pp. 44-47, 2008.

[5] A. Kharb, V. Saini, Y. K. Jain e S. Dhiman, «A review of gait cycle and its parameters.,» 2011. [Online]. Available: www.IJCEM.orgIJCEMwww.ijcem.org.

[6] D. Sethi, S. Bharti e C. Prakash, «A comprehensive survey on gait analysis: History, parameters, approaches, pose estimation, and future work,» *Artificial Intelligence in Medicine,* vol. 129, 2022.

[7] J. Perry e M. G. Benedetti, Analisi del movimento, 1995.

[8] W. Pirker e R. Katzenschlager, «Gait disorders in adults and the elderly: A clinical guide,» *Wiener klinische Wochenschrift,* vol. 129, n. 3, 2016.

[9] F. Buganè, M. G. Benedetti, G. Casadio, S. Attala, F. Biagi, M. Manca e A. Leardini, «Estimation of spatial-temporal gait parameters in level walking based on a single accelerometer: Validation on normal subjects by standard

gait analysis,» *Computer Methods and Programs in Biomedicine,* vol. 108, n. 1, pp. 129-137, 2012.

[10] E. S. Grood e W. J. Suntay, «A joint coordinate system for the clinical description of three-dimensional motions: Application to the knee,» *Journal of Biomechanical Engineering,* vol. 105, n. 2, pp. 136-144, 1983.

[11] G. Wu, F. C. T. van der Helm, H. E. J. Veeger, M. Makhsous, P. Van Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, F. W. Werner e B. Buchholz, «ISB recommendation on definitions of joint coordinates systems of various joints for the reporting of human joint motion - Part II: shoulder, elbow, wrist, and hand,» *Journal of Biomechanics,* vol. 38, n. 5, pp. 981-992, 2005.

[12] G. Wu, S. Siegler, P. Allard, C. Kirtley, A. Leardini, D. Rosenbaum, M. Whittle, D. D. D'Lima, L. Cristofolini, H. Witte, O. Schmid e I. Stokes, «ISB recommendation on definitions of joint coordinates systems of various joints for the reporting of human joint motion - Part I: ankle, hip, and spine,» *Journal of Biomechanics,* vol. 35, n. 3, pp. 543-548, 2002.

[13] M. Bertoli, A. Cereatti, D. Trojaniello, L. Avanzino, E. Pelosin, S. Del Din, L. Rochester, P. Ginis, E. M. J. Bekkers, A. Mirelman, J. M. Hausdorff e U. Della Croce , «Estimation of spatio-temporal parameters of gait from magneto-inertial measurment units: multicenter validation among Parkinson, midly cognitively impaired and healthy older adults,» *Biomedical Engineering Online,* vol. 17, n. 58, 2018.

[14] M. H. Khan, M. S. Farid e M. Grzegorzek, «Vision-based approaches towards person identification using gait,» *Computer Science Review,* vol. 42, 2021.

[15] A. Cappozzo, U. Della Croce, A. Leardini e L. Chiari, «Human movement analysis using stereophotogrammetry Part 1: theoretical background,» *Gait and Posture,* vol. 21, pp. 186-196, 2005.

[16] L. Wade, L. Needham, P. McGuigan e J. Bilzon, «Applications and limitations of current markerless motion capture methods for clinical gait biomechanics.,» *PeerJ,* 2022.

[17] J. BeomJun e K. SeongKi, «Comparative Analysis of OpenPose, PoseNet, and MoveNet Models for Pose Estimation in Mobile Devices,» *International Information and Engineering Technology Association,* vol. 39, n. 1, pp. 119-124, 2022.

[18] V. Rani e M. Kumar, «Human gait recognition: A systematic review,» *Multimedia Tools and Applications,* vol. 82, pp. 37003-37037, 2023.

[19] K. A. Lamkin-Kennard e M. B. Popovic, «Sensors: Natural and Synthetic Sensors,» *Biomechatronics,* pp. 81-107, 2019.

[20] F. Salis, S. Bertuletti, T. Bonci, M. Caruso , K. Scott, L. Alcock, E. Buckley, E. Gazit, C. Hansen, L. Schwickert , K. Aminian , C. Becker , P. Brown, A. Carsin, B. Caulfield, L. Chiari, I. D'Ascanio, S. Del Din, B. Eskofier, J. Garcia-Aymerich, J. Hausdorff , E. Hume , C. Kirk , F. Kluge , S. Koch, A. Kuederle, W. Maetzler , E. Micó-Amigo, A. Mueller , I. Neatrour , A. Paraschiv-Ionescu , L. Palmerini, A. Yarnall, L. Rochester, B. Sharrack, D. Singleton, B. Vereijken , I. Vogiatzis, U. Della Croce, C. Mazzà e A. Cereatti, «A multi-sensor wearable system for the assessment of diseased gait in real-world conditions.,» *Frontiers in Bioengineering and Biotechnology,* vol. 11, 2023.

[21] A. M. Sabatini, «Estimating Three-Dimensional Orientation of Human Body Parts by Inertial/Magnetic Sensing,» *Sensors,* vol. 11, n. 2, pp. 1489-1525, 2011.

[22] S. M. Alfayeed e B. S. Saini, «Human Gait Analysis Using Machine Learning: A Review,» *International Conference on Computational Intelligence and Knowledge Economy (ICCIKE),* 2021.

[23] A. Kececi, A. Yildirak, K. Ozyazici, G. Ayluctarhan e O. Agbulut, «Implementation of machine learning algorithms for gait recognition,» *Engineering Science and Technology, an International Journal,* vol. 23, pp. 931-937, 2020.

[24] J. E. Boyd e J. J. Little, «Biometric Gait Recognition,» *Advanced Studies in Biometrics, Springer,* pp. 19-42, 2005.

[25] M. H. Khan, M. S. Farid e M. Grzegorzek, «Person identification using spatiotemporal motion characteristics,» *Proceedings of International Conference on Image Processing,* pp. 166-170, 2017.

[26] A. Nambiar, A. Bernardino e J. C. Nascimento, «Gait-based Person Re-identification: A Survey,» *ACM Computing Surveys,* vol. 52, n. 2, 2019.

[27] A. Kale, A. Sundaresan, A. N. Rajagopalan, N. P. Cuntoor, A. K. Roy-Chowdhury, V. Kruger e R. Chellappa, «Identification of Human Using Gait,» *IEEE Transaction on Image Processing,* vol. 13, n. 9, pp. 1163-1173, 2004.

[28] M. P. Murray, A. B. Drought e R. C. Kory, «Walking patterns of normal men,» *Journal of Bone and Joint Surgery,* Vol. %1 di %246-A, n. 2, pp. 335-360, 1964.

[29] M. P. Murray, «Gait as a total pattern of movement,» *American Journal of Physical Medicine & Rehabilitation,* vol. 46, pp. 290-332, 1967.

[30] J. P. Singh, S. Jain, S. Arora e U. P. Singh, «Vision-based Gait Recognition: A Survey,» *IEEE Access,* vol. 6, pp. 70497-70527, 2018.

[31] C. Wan, L. Wang e V. V. Phoha, «A Survey on Gait Recognition,» *ACM Computing Surveys,* vol. 51, n. 5, 2018.

[32] C. BenAbdelkader, R. Cutler e L. Davis, «Stride and cadence as a biometric in automatic person identification and verification,» *Proceedings of 5th IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 372-377, 2002.

[33] I. Bouchrika e M. S. Nixon, «Model-based feature extraction for gait analysis and recognition,» *International Conference on Computer Vision,* pp. 150-160, 2007.

[34] F. Tafazzoli e R. Safabakhsh, «Model-based human gait recognition using leg and arm movements,» *Engineering Applications of Artificial Intelligence,* vol. 23, pp. 1237-1246, 2010.

[35] J. H. Yoo, D. Hwang, K. Y. Moon e M. S. Nixon, «Automated human recognition by gait using neural network,» *Image Processing Theory, Tools and Applications,* pp. 1-6, 2008.

[36] Y. Wang, J. Sun, J. Li e D. Zhao, «Gait recognition based on 3D skeleton joints captured by Kinect,» *IEEE International Conference on Image Processing,* pp. 3151-3155, 2016.

[37] G. Zhao, G. Liu, H. Li e M. Pietikainen, «3D gait recognition using multiple cameras,» *7th International Conference on Automatic Face and Gesture Recognition,* pp. 529-534, 2006.

[38] R. Liao, Z. Li, S. Bhattacharyya e G. York, «Posemapgait: A model-based gait recognition method with pose estimation maps and graph convolutional networks,» *Neurocomputing,* vol. 501, pp. 514-528, 2022.

[39] A. S. M. H. Bari e M. L. Gravilova, «Artificial Neural Network Based Gait Recognition Using Kinect Sensor,» *IEEE Access,* vol. 7, 2019.

[40] A. F. Bobick e J. W. Davis, «The recognition of human movement using temporal templates,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 257-267, 2001.

[41] J. Han e B. Bhanu, «Individual recognition using gait energy image,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, pp. 316-322, 2006.

[42] J. Liu e N. Zheng, «Gait history image: A novel temporal template for gait recognition,» *IEEE International Conference on Multimedia and Expo,* pp. 663-666, 2007.

[43] A. Kale, A. N. Rajagopalan, N. Cuntoor e V. Kruger, «Gait-based recognition of humans using continuous HMMs,» *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 336-341, 2002.

[44] A. Sheth, M. Sharath, S. C. Reddy e K. Sindhu, «Gait Recognition Using Convolutional Neural Networks,» *International Journal of Online and Biomedical Engineering,* vol. 19, n. 1, pp. 107-118, 2023.

[45] W. A. Alsaggaf, I. Mehmood, E. F. Khairullah, S. Alhuraiji, M. Sabir, A. S. Alghamdi e A. A. A. El-Latif, «A smart surveillance system for uncooperative gait recognition using cycle consistent generative adversarial networks (CCGANs),» *Computational Intelligence and Neuroscience,* 2021.

[46] H. Guo, B. Li, Y. Zhang, Y. Zhang, W. Li, F. Qiao, X. Rong e S. Zhou, «Gait Recognition Based on the Feature Extraction of Gabor Filter and Linear Discriminant Analysis and Improved Local Coupled Extreme Learning Machine,» *Mathematical Problems in Engineering,* 2020.

[47] Microsoft, «Azure Kinect DK hardware specifications,» [Online]. Available: https://learn.microsoft.com/en-us/azure/kinect-dk/hardware-specification. [Consultato il giorno 25 August 2024].

[48] V. VUE, «VUE Ready set mocap.,» [Online]. Available: http://www.helmar-ms.pl/helmar-ms/plik/vicon_vue_nn4672.pdf. [Consultato il giorno 25 August 2024].

[49] M. Li, Z. Zhou, J. Li e X. Liu, «Bottom-up Pose Estimation of Multiple Person with Bounding Box Constraint,» *Computer Vision and Pattern Recognition,* 2018.

[50] «COCO - Common Objects in Context,» [Online]. Available: https://cocodataset.org/#home. [Consultato il giorno 18 09 2024].

[51] R. Votel e L. Na, «Next-Generation Pose Detection with MoveNet and TensorFlow.js,» 17 May 2021. [Online]. Available: https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html. [Consultato il giorno 18 September 2024].

[52] J. BeomJun e K. SeongKi, «Comparative Analysis of OpenPose, and MoveNet Models for Pose Estimation in Mobile Devices,» *International Information and Engineering Technology Association,* vol. 39, n. 1, pp. 119-124, 2022.

[53] I. Serouis, 2022. [Online]. Available: https://www.kaggle.com/code/ibrahimserouis99/human-pose-estimation-with-movenet. [Consultato il giorno 19 July 2024].

[54] D. Balta, G. Figari, G. Paolini, E. Pantzar-Castilla, J. Riad, U. Della Croce e A. Cereatti, «A Model-Based Markeless Protocol for Clinical Gait Analysis Based on a Single RGB-Depth Camera: Concurrent Validation on Patients With Cerebral Palsy,» *Access,* vol. 11, pp. 144377-144393, 2023.

[55] M. Roberts, D. Mongeon e F. Prince, «Biomechanical parameters for gait analysis: a systematic review of healthy human gait,» *Physical Therapy and Rehabilitation,* vol. 4, n. 6, pp. 2055-2386, 2017.

[56] M. G. Benedetti, F. Catani, A. Leardini, E. Pignotti e S. Giannini, «Data management in gait analysis for clinical applications,» *ELSEVIER,* pp. 204-215, 1998.

[57] C. Ferraris, G. Amprimo, G. Masi, L. Vismara, R. Cremascoli, S. Sinagra, G. Pettiti, A. Mauro e L. Priano, «Evaluation of Arm Swing Features and

Asymmetry during Gait in Parkinson's Disease Using Azure Kinect Sensor,» *Sensors,* August 2022.

[58] M. Blazkiewicz, I. Wiszomirska e A. Wit, «Comparison of four methods of calculating the symmetry of spatial-temporal parameters of gait,» *Acta of Bioengineering and Biomechanics,* vol. 16, n. 1, 2014.

[59] C. Buckley, B. Galna, L. Rochester e C. Mazzà, «Attenuation of Upper Body Accelerations during Gait: Piloting an Innovative Assessment Tool for Parkinson's Disease,» *BioMed Research International,* pp. 2-4, 8, 9, 2015.

[60] Y. Saeys, I. Inza e P. Larrañaga, «A review of feature selection techniques in bioinformatics,» *Bioinformatics,* vol. 23, n. 19, pp. 2507-2517, 2007.

[61] Y. Li e S. Abdallah, «On hyperparameter optimization of machine learning algorithms: Theory and practice,» *Neurocomputing,* vol. 415, pp. 295-316, July 2020.

[62] M. Grandini, E. Bagli e G. Visani, «Metrics for Multi-Class Classification: An Overview,» 14 August 2020. [Online]. Available: https://arxiv.org/pdf/2008.05756. [Consultato il giorno 22 August 2024].