Direct Estimation Of The Knee Flexion Angle From A Monocular Image During Walking

by

Anis FAKHFAKH

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER’S DEGREE
WITH THESIS IN INFORMATION TECHNOLOGY
M.A.Sc.

MONTREAL, AUGUST 18, 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

Anis FAKHFAKH, 2025

This Creative Commons license allows readers to download this work and share it with others as long as the author is credited. The content of this work cannot be modified in any way or used commercially.

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Carlos Vázquez, Thesis supervisor
Department of Software Engineering and IT, École de technologie supérieure

Ms. Nicola Hagemeister, Chair, Board of Examiners
Department of System Engineering, École de technologie supérieure

Ms. Neila Mezghani, External Examiner
Department of Science and Technology, Université TÉLUQ

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON AUGUST 15, 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

This thesis marks the culmination of a significant chapter in my academic journey, and I am deeply grateful to those who have supported and guided me along the way.

Foremost, I would like to express my sincere thanks to my supervisor, Prof.Carlos Vázquez and Dr. Thierry Cresson for their invaluable guidance, support, and feedback throughout the development of this work. Their expertise and encouragement were instrumental in helping me overcome the technical challenges of this research.

I would also like to extend my gratitude to the members of the Laboratoire d'imagerie et d'orthopédie (LIO) at ÉTS and the Centre de recherche du CHUM (CRCHUM) for providing a favorable research environment and access to essential tools and data.

A special thank you goes to the members Emovi and Moveck for their partnership and commitment in making this project possible by granting access to the necessary tools and related resources. Their commitment to advancing research in biomechanics is truly appreciated.

On a more personal note, I am profoundly grateful to my family and close friends for their unwavering support, patience, and encouragement throughout this journey. Their belief in me helped me stay focused and determined during difficult moments and maintain momentum.

Lastly, I would also want to thank all those who volunteered to take part of the data collection protocol. Without your collaboration, this project wouldn't succeed.

Estimation directe de l’angle de flexion du genou pendant la marche à partir d’une image RGB

Anis FAKHFAKH

RÉSUMÉ

L’utilisation récente de techniques d’apprentissage profond dans les applications biomécaniques a ouvert de nouvelles opportunités pour la capture de mouvement et l’évaluation fonctionnelle à partir d’images, de manière non-invasive. Avec l’augmentation de l’importance des modèles de vision par ordinateur basés sur l’apprentissage profond, l’entreprise Emovi vise à remplacer les méthodes de capture de mouvement basées sur le système KneeKG (capture de mouvement avec marqueurs) par des stratégies sans marqueurs permettant d’estimer et d’évaluer certains paramètres anatomiques des membres inférieurs.

Ce mémoire propose un nouveau modèle de régression directe, construit autour d’une architecture de base ResNet et enrichi par des contraintes sur les cartes de caractéristiques intermédiaires, afin d’estimer directement les angles de flexion du genou à partir des séquences de mouvements fonctionnels sous forme d’images RGB monoculaires. Cette supervision intermédiaire sur les cartes de caractéristiques vise à renforcer la cohérence spatiale des représentations internes du réseau.

Une base de données a été acquise dans un environnement de laboratoire contrôlé en utilisant le système KneeKG, qui fournit des données cinématiques de référence validées cliniquement. Contrairement aux approches antérieures basées sur des configurations multi-vues ou des systèmes complexes avec marqueurs, ce travail introduit une méthode capable d’estimer les angles articulaires du genou à partir d’une seule image monoculaire.

Ce travail introduit également des stratégies d’inpainting d’image afin d’encourager un apprentissage du modèle plus orientévers la tâche d’estimation. Les évaluations quantitatives démontrent une erreur absolue moyenne de 2.4° sur les données de test pour l’estimation de l’angle de flexion. Ces résultats indiquent un potentiel d’application en contexte clinique où des outils d’analyse du mouvement accessibles et à faible coût sont favorisés.

Mots-clés: Vision par ordinateur, Apprentissage profond, MoCap, Flexion, KneeKG, Cartes de caractéristiques, Inpainting d’image

Direct Estimation Of The Knee Flexion Angle From A Monocular Image During Walking

Anis FAKHFAKH

ABSTRACT

The recent use of deep learning techniques in biomechanical applications has opened new avenues for non-invasive image-based motion capture and assessment. With the growing importance of deep learning-based computer vision models, the healthcare innovation company Emovi aims to replace KneeKG-based MoCap Marker-based methods with markerless-based strategies to estimate and assess certain anatomical landmarks of the lower limbs.

This thesis proposes a novel deep regression model built with a ResNet backbone enhanced with feature map constraints to directly estimate lower limb anatomical landmarks, specifically knee flexion angles; by enforcing spatial consistency through internal feature map supervision; from monocular RGB images of defined functional movement sequences.
A dataset was collected in a controlled laboratory environment using the KneeKG system, which provides clinically validated kinematic reference data. Unlike prior methods that rely on multi-view setups or complex marker-based systems, this work introduces an approach to infer knee joint angles from a RGB image.

This works also introduces image inpainting strategies to help with a more focused model training. Quantitative evaluations demonstrate a test mean absolute error of 2.4° in flexion angle estimation, indicating a potential use for clinical applications where cost-effective and accessible motion analysis tools are needed.

Keywords: Computer Vision, Deep Learning, MoCap, Flexion, KneeKG, Feature maps, Image inpainting

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ETS	École de Technologie Supérieure
CV	Cross-validation
HPE	Human pose estimation
MSE	Mean squared error
MAE	Mean absolute error
AE	Absolute error
SPM	Statistical parametric mapping
SPM1D	1D-Statistical parametric mapping
SOTA	State-of-the art
NGC	Normalized gait cycle
BB	Bounding box

INTRODUCTION

0.1 Context: KneeKG and Motion capture systems

Motion analysis plays a fundamental role in the fields of biomechanics, clinical diagnosis (muscle activity, movement patterns), rehabilitation (post-injury recovery and physical therapy) and sports (Winter, 2009). In biomechanics, motion analysis allows to assess important details like the joint kinematics, which are essential for monitoring the knee state, identifying abnormal movement patterns, risks and signs of injuries and enhancing the performance of high-level athletes (Zong-Hao,MA, Li & He, 2023).

KneeKG^{TM}, a Knee Kinesiography device to assess joint motion, is part of a marker-based motion capture system that was designed to better identify the causes of the symptoms and to analyze the movement by identifying the joints using cinematic landmarks. KneeKG helps in identifying symptoms of Knee osteoarthritis (OA); which is a degenerative joint disease (Hsu & Siwiec, 2023) by monitoring the knee condition. Its development started in 1992 with the Laboratoire d'imagerie et d'orthopédie (LIO) in Montreal. LIO and Emovi collaborated on developing a knee medical exam based on the KneeKG tool and its accessories and consisting of a series of experiments executed in a laboratory-constraint environment with the KneeKG tool and with the presence of dedicated clinicians or trained professionals. Thanks to this exam, it is possible to extract the 3D knee kinematics by performing a knee exam on a treadmill and with the help of dedicated material (infrared camera and reflective markers). With the commercialization of the product, it became easy to process biomechanical examination reports (Lustig, Magnussen, Cheze & Neyret, 2012; Hagemeister *et al.*, 2005)

More generally, according to Wade, Needham, McGuigan & Bilzon (2022), marker-based motion capture involves attaching reflective markers to the skin and using close range infrared cameras to precisely track movement. Some of the most common identified anatomical landmarks are the joint positions and angles in three dimensions, describing body segment poses. The markers are detected with high accuracy, allowing for the determination of body segment locations and orientations. The marker-based motion capture technology represents the poses in the form of six degrees of freedom: three translational ("sagittal, frontal, and transverse position") and three rotational ("flexion/extension, abduction/adduction, and rotation about the longitudinal axis").

While being the gold standard in terms of produced data, marker-based technologies in general have significant limitations. The need for a controlled environment can affect how participants move naturally due to the Hawthorne effect and being under observation (McCambridge, Witton & Elbourne, 2013). Baker (2006); Wade *et al.* (2022); Antognini *et al.* (2025) state that the cost of equipment and the requirement for highly trained personnel with controlled laboratory in-person intervention make it less feasible for many clinical applications. Additionally, other factors such as the time cost for marker placement, the limited capture systems reach and occlusion issues make the data less accurate and reliable and thus making the marker-based systems less practical than markerless systems for widespread or dynamic use. The marker placement is also prone to uncertainty that can be caused by either human error or simply some soft tissue artifacts (STA) caused by skin, muscle, and fat movement (Antognini *et al.*, 2025). For instance, Fiorentino *et al.* (2017) stated that STA causes substantial joint angle errors that vary according to the static or dynamic factors.
In particular, the limitations of KneeKG are the need of having a clinician on-site carrying the attached device and performing the anatomical calibration; making it impossible to generate cinematic parameters (kinematics) provided by KneeKG when it is inaccessible for certain patients or health institutions. This underscores the necessity of a remote-capable solution that can accurately estimate kinematic parameters, enabling efficient telemedicine-based assessments. The motion capture systems that are not based on reflective markers for motion tracking are called markerless motion capture systems.

0.2 Markerless motion capture

Markerless Motion Capture (MMC) often relies on video footage recorded from a single or multi-camera setup to be processed through a deep learning-based pose estimation algorithm. Markerless systems use either depth or standard video cameras (Wade *et al.*, 2022). Markerless motion capture systems based on depth cameras (e.g., Microsoft Azure Kinect) capture video and depth data but have notable limitations and discrepancies compared to marker-based systems: Mentiplay *et al.* (2015) stated that Microsoft Kinect V2 is not able to accurately assess lower body kinematics during gait. Rodrigues *et al.* (2019) compared their own developed multimodal system including multiple RGB-D cameras and multiple IMU sensors with the VICON system and cited that even though it is an accurate and reliable motion analysis system, it still shows uncertainties compared to marker-based systems.

The use of standard RGB video hardware for markerless motion capture purposes is still limited compared to depth-based hardware by the position, number of cameras and the lightning settings but are practical to collect long-range data through zoom lenses during sporting activities (for example Hawk-Eye Tennis Tracking system (Owens, Harris & Stennett, 2003)). It remains a challenging task to accurately extract more detailed information like joint centers from recorded footage (Wade *et al.*, 2022). For that purpose, Human pose estimation (HPE) methods have emerged as a markerless approach to automate the estimation of the human pose and specifically the lower limbs pose.

Human pose estimation

In its commonly used meaning, HPE involves detecting and localizing human body keypoints (joints like elbows, knees) from images or videos, enabling applications in action recognition, human-computer interaction, and healthcare (Knap, 2024). Joint-based pose estimation algorithms predict joint coordinates from image/video-based datasets of different sources. The annotation techniques of these datasets vary according to the use case. Some datasets are either manually annotated (COCO pose dataset (Lin *et al.*, 2014), FLIC dataset (Johnson & Everingham, 2010), etc.) or annotated using MoCap systems: Human3.6M (Ionescu, Papava, Olaru & Sminchisescu, 2014) for example is annotated using reflective markers and infrared cameras). With the emergence of Deep Learning, the first neural network to train on pose estimation datasets (Toshev & Szegedy, 2014) was developed allowing for further work to improve the accuracy of the predictions (Tompson, Jain, LeCun & Bregler, 2014; Newell, Yang & Deng, 2016; Cao, Simon, Wei & Sheikh, 2017; He, Gkioxari, Dollár & Girshick, 2017; Tripathi, Ranade, Tyagi & Agrawal, 2020; Martinez, Hossain, Romero & Little, 2017).

With advancements in state-of-the-art (SOTA) pose estimation, the markerless estimation of kinematic parameters became increasingly accessible and reliable. In particular, it is now possible to estimate anatomical angles such as the knee joint flexion/extension angle without the need for physical markers. Deep learning-based approaches are increasingly used for this purpose. Their performance is typically assessed by comparing the predicted joint angles to the ground-truth values captured from the marker-based motion capture systems (Stenum *et al.*, 2021; Kidzinski *et al.*, 2020; Le & Pham, 2024; Dinh *et al.*, 2025; Kumar *et al.*, 2021; Jamsrandorj, Kumar, Arshad, Mun & Kim, 2022).

However, the reliability of the HPE methods in a clinical context depends on the performance of these models on real clinical data acquired in controlled environments. While some joint-based approaches are trained and tested on public benchmark datasets that are manually annotated, the error values and accuracy might be different for a dataset acquired in a clinical context. In biomechanics, a flexion angle error is considered to be acceptable if it lies within the range between 2 and 5 degrees. In fact, McGinley, Baker, Wolfe & Morris (2009) mention that the 2 to 5 degrees range is a reasonable error rate for estimating the flexion angles with markerless systems compared to marker-based systems obtained angles and that the error is highly likely to be clinically acceptable if it is inferior to 2 degrees.

In the context of this project, EMOVI uses the same limits for the KneeKG system to determine the acceptance of the markerless estimations: if the error is inferior to 2 degrees, the markerless system performance is considered optimal; and if the error is between 2 and 5 degrees, the performance is considered reasonable.

Since the HPE datasets in clinical context are acquired using marker-based motion capture systems, the data source (images and videos) will potentially include reflective markers that might bias the evaluation of the markerless pose estimation method ability at making accurate predictions. It is important to adapt the pose estimation method to generalize on images without the attached markers.

0.3 Objectives of the project

Based on the motivations of this introduction, the present work aims to develop an automated method to estimate the joint angles, specifically the knee flexion angles, using deep learning techniques applied to monocular RGB images. The method is expected to produce clinically accurate results (McGinley *et al.*, 2009).

CHAPTER 1

LITERATURE REVIEW

In this chapter, we first review the SOTA in HPE. We then introduce the fundamental deep learning concepts relevant to our study. Finally, we explore specific data processing techniques along with the methodology used for validating the approach and evaluating its performance on the dataset.

1.1 Human pose estimation

To have an understanding of the existing research landscape and identify techniques relevant to the core objective of this study, we review the deep learning literature related to HPE in this Section 1.1. The definition of the pose varies according to the context and the studied object/subject. We explore 3 different pose definitions and detail the associated SOTA methods.

1.1.1 Keypoint-based Human Pose Estimation

HPE is the estimation of the posture of a person from an input source (image, video, etc.) (Chen, Tian & He, 2020). According to Zheng *et al.* (2023), the human body can be seen as a set of joints and limbs oriented according to a certain scheme. To represent the pose, keypoints and key features are extracted from the input. This action requires an appropriate representation of the human body according to the pose estimation model used. The recent research in deep learning is focused on kinematic models (also known as skeleton-based models). It is represented as a set of joint locations and the corresponding limb orientations following the human body skeletal structure. It can be seen as a graph where vertices indicate the joints and the edges encode the constraints or prior connections of joints. While this model is very simple, flexible to represent different poses with a minimalist set of variables and parameters and is well suited to estimate both 2D and 3D poses, it is limited in texture and shape representation (Chen *et al.*, 2020). Hence, this subSection 1.1.1 will focus on this representation as it is suitable for the marker-based data acquisition of this project: it offers the flexibility of inferring angles from joints.

The two main options to separate HPE deep learning models are single/multi-person based or 2D/3D-based. Single person HPE methods are designed to estimate the pose of only one person in the image while multi-person HPE methods do it for multiple persons.
Since the case study has one subject per trial, this review will focus mainly on single person methods but multi-person approaches are still viable and applicable. Specifically, top-down approaches first detect the bounding boxes (BBs) of the persons present in the image then predict the joint coordinates of each person individually in their corresponding BB. The coordinates are then scaled back to match the original image (Wang, Zhang & Ge, 2021). The top-down approaches are basically a combination of person's BB detector (like RCNN (Girshick, Donahue, Darrell & Malik, 2014) or Faster RCNN (Ren, He, Girshick & Sun, 2017)) followed by a group of single person pose estimators for each detected person. Their joint estimators are trained on individual persons BBs. Later on inference, the pipeline infers the pose for each detected person. Some notable methods are Deep High-Resolution Representation Learning for HPE (Sun *et al.*, 2019a) HRNet and SimpleBaseline (Xiao, Wu & Wei, 2018).

According to Dubey & Dixit (2023), the 3D pose estimation consists of predicting an extra layer of the information in the pose space: the depth. This task has multiple constraints to consider during learning such as textures and skin color variations, occlusions, perturbations in human background, etc.
Compared to 2D HPE, the 3D HPE poses several challenges. Zheng *et al.* (2023) state that 3D estimation requires learning an extra layer of knowledge related to depth which is not always straightforward. The other challenge is the lack of data that might be relevant because 3D pose estimation models are less likely to generalize on different datasets. There is a limited amount of public 3D human pose datasets; which can be explained by the effort it takes to generate such datasets under strict experimental instructions and expensive equipment (Martinez *et al.*, 2017). 3D pose datasets like Human3.6M (Ionescu *et al.*, 2014) are obtained with a dedicated marker-based motion capture system with cameras in a controlled laboratory environment. Since our dataset does not have a diversity in depth information, estimating 3D information will only make the learning task more complex.

Before the deep learning models breakthrough, HPE pipelines were mostly reliant on generative models based on prior learning, likelihood optimization, etc. encoding a parametric model which learns a certain prior to generate the body parts based on a graphical inference to represent the spatial relationships between the body parts (Josyula & Ostadabbas, 2021). Discriminative models (essentially ML/DL-based) learn a direct mapping from the input image/video source to the pose space (joint coordinates) making them faster and more convenient that the generative models (Chen *et al.*, 2020).

Conventionally, the 2D single person pose estimation models (Toshev & Szegedy, 2014; LI, Liu & Chan, 2014; Wei, Ramakrishna, Kanade & Sheikh, 2016; Newell *et al.*, 2016; Ke, Chang, Qi & Lyu, 2018) include a BB person detector to crop the image of the person of interest into a sub-image. Generally, the source (image/video) contains just one person but in case of the presence of multiple persons, the detection selects the most likely person to estimate its pose. The BB detectors can either be full-body or upper-body detectors depending on the pose estimation task and dataset: BBC pose ((Charles, Pfister, Everingham & Zisserman, 2014)) and FLIC (Sapp & Taskar, 2013) datasets have upper-body annotated datasets but Human3.6M (Ionescu *et al.*, 2014) and Leeds Sports Pose (LSP) Johnson & Everingham (2010) datasets have full-body annotations.

For single person HPE, there are 2 types of methods (Zheng *et al.*, 2023):

- Regression methods apply an end-to-end framework to directly learn a mapping from the input image to body joints or parameters of human body models.
- Heatmap-based methods predict approximate locations of body parts and joints by using supervised representative heatmaps and image patches.

Regression-based methods are more direct and require less supervision from intermediate feature maps than heatmap-based ones, but they lack flexibility in terms of the non-linearity of the estimation problem (Chen *et al.*, 2020). The use of pretrained feature extractors is very common in literature since they are trained on a large number of image datasets with a rich feature space. They are either loaded with their pretrained weights without further retraining or they are refined on the pose image datasets. The features maps from the feature extractors are then sent to the regression module to predict the keypoints. Toshev & Szegedy (2014) used AlexNet (Krizhevsky, Sutskever & Hinton, 2012) followed by a cascade of predictors to refine the pose estimation by adjusting small displacements in the initial pose estimations. Sun, Shang, Liang & Wei (2017) implemented Compositional Human Pose: a Structure-aware regression module based on a bone-specific representation of the human body instead of the traditional joint scheme using the ImageNet-pretrained ResNet-50 (He *et al.*, 2016) backbone. Using double cascaded encoder-decoder self-attention transformers, "Pose Recognition With Cascade Transformers" (Li *et al.*, 2021) model employs the first transformer to detect the person and the second one for the joints coordinates regression. LI *et al.* (2014) uses a heterogeneous multi-task DNN framework consisting of two tasks: joints coordinates regression from full images and body parts detection from image patches. The shared feature representation allows for a better performance on both tasks.

Heatmap-based approaches are widely used in 2D HPE because they provide a robust way to represent joint locations as probability distributions, allowing the model to precisely locate the joints. These methods predict heatmaps where each joint keypoint is represented by a 2D Gaussian distribution centered at its true location (Zheng *et al.*, 2023). Convolutional pose machines by Wei *et al.* (2016) rely on a sequential structure composed of convnets for multi-stage refinement using the supervision of intermediate heatmaps. The prediction from each stage is a refinement from the heatmap (output) obtained from the previous stage. Similarly, Stacked Hourglass Networks for HPE (Newell *et al.*, 2016) used a multi-stage refinement strategy by sequentially stacking multiple hourglass modules (where each module output is used to refine the next one) and using intermediate supervision. Multi-Scale Structure-Aware Network for Human Pose Estimation (Ke *et al.*, 2018) is an approach that is based on multi-scale supervision and multi-scale regression enhanced by a joint body model S. Belagiannis & Zisserman (2017) iteratively refines the heatmaps predictions with a single recurrent architecture instead of cascaded architectures. Adversarial learning is also used in some approaches like adversarial PoseNet (Chen, Shen, Wei, Liu & Yang, 2017): a structure-aware conditional adversarial network containing an hourglass network called the generator and two discriminators; one to discriminate against unrealistic poses and the other one to discriminate against predictions with weak confidence in locating the body parts.

Deep High-Resolution Representation Learning for Human Pose Estimation

Typical convolutional architectures for pose estimation follow a high-to-low-resolution pipeline (like Newell *et al.* (2016)) followed by an upsampling pipeline to recover the spatial information. To preserve fine spatial information, HRNet maintains high-resolution representation through the network without an irreversible loss of important information by adding parallel streams at progressively lower resolutions.

HRNet is based on two fundamental parts: parallel multi-resolution subnetworks where the network starts with a high-resolution branch and progressively adds parallel ones for lower resolutions as it goes deeper. The parallel structure is complemented by a multi-scale fusion at various stages of the network where the features are exchanged and aggregated between the parallel stems in both directions (high to lower and low to higher).
High to lower fusions are typically achieved with strided convolutions and low to higher fusions rely on a 1×1 convolution followed by the nearest neighbor up-sampling.
The exchange and fusion are done through four stages with four parallel subnetworks with a gradually decreased output resolution. The continuous exchange from different resolutions allows the network to maintain a rich representation capable of estimating a precise pose.

The final loss function minimizes the discrepancy between the ground truth heatmaps and the predicted ones from the main pose estimation head relative to the highest resolution feature map. The ground-truth heatmaps are generated by creating gaussian blurred representations around the keypoint locations. The model predicts a number of heatmaps equal to the number of joints where for each map, the coordinates of the pixel with the highest value are the final predicted keypoint coordinates.

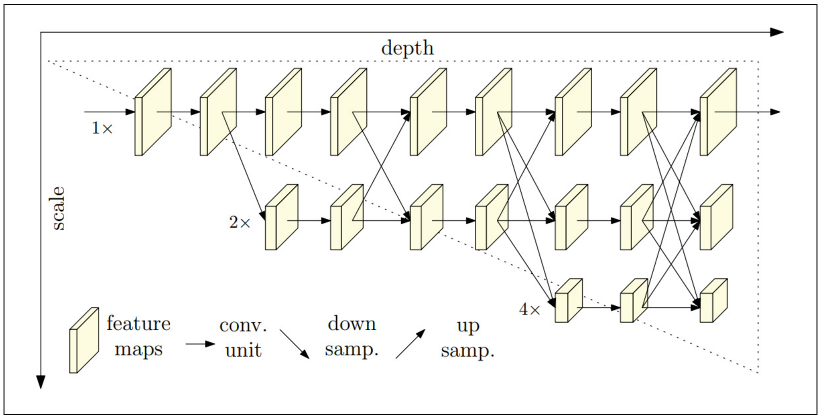


Figure 1.1 General architecture of HRNet
Taken from Sun *et al.* (2019a)

1.1.2 Orientation-based pose estimation

Contrary to keypoint-based pose estimation that estimates coordinates-based information, pose orientation in this context is the rotation orientation of an object with respect to another reference (for example the global axis system). The poses are based on some conventional representations. They define how an object or body is rotated in a three-dimensional space and are crucial for tasks such as 3D HPE, motion capture, robotic control, and augmented reality applications. Each representation has its own applications and limitations making the choice of representation impactful on the accuracy of the pose estimation and also the numerical stability and learning efficiency of machine learning models for learning certain patterns. The mathematical properties of the standard rotation spaces show they have non-euclidean properties in a special orthogonal space $SO(3)$ (Grassia, 1998; Zhou, Barnes, Lu, Yang & Li, 2019).

Euler angles

Euler angles represent rotations by decomposing them into three sequential rotations of angles θ_1, θ_2 and θ_3 around the coordinate axes (XYZ, ZYX, etc.) (McIntosh, 2019).
For example, the XYZ representation is a series of three consecutive rotation angles (θ_X, θ_Y and θ_Z) respectively around the three axes X, Y and Z.

Despite being compact and intuitive, Euler angles representations suffer from gimbal lock, a situation where two of the three rotation axes align, resulting in a loss of one degree of freedom (McIntosh, 2019) and from space discontinuity (Zhou *et al.*, 2019).

Rotation matrix representation

Rotation matrices (3×3 orthonormal matrices) are a straightforward and unambiguous representation of rotation. However, enforcing orthogonality and the determinant to be equal to 1 during learning can be challenging (Zhou *et al.*, 2019). It also faces the problem of space discontinuity (Peretroukhin *et al.*, 2020) requiring an orthogonality constraint on the nine parameters of the matrix by Gram-Schmidt (GS) process (Zhou *et al.*, 2019).

There is a straightforward equivalence between Euler angles and rotation matrices representations. In fact, for a given sequence (XYZ for example):

The rotation about the X-axis by angle θ_X is defined as

$$R_x(\theta_X) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_X & -\sin\theta_X \\ 0 & \sin\theta_X & \cos\theta_X \end{bmatrix} \tag{1.1}$$

The rotation about the Y-axis by angle θ_Y is defined as

$$R_y(\theta_Y) = \begin{bmatrix} \cos\theta_Y & 0 & \sin\theta_Y \\ 0 & 1 & 0 \\ -\sin\theta_Y & 0 & \cos\theta_Y \end{bmatrix} \tag{1.2}$$

The rotation about the Z-axis by angle θ_Z is defined as

$$R_z(\theta_Z) = \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & 0 \\ \sin\theta_Z & \cos\theta_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

The combined rotation matrix R will have the form:

$$R = R_z(\theta_Z)R_y(\theta_Y)R_x(\theta_X) \tag{1.4}$$

The inverse transformation from a rotation matrix R to a series of Euler angles (according to a defined Euler sequence) is also possible and the formula differs according to the sequence (Diebel, 2006).

Axis-angle representation

The axis-angle representation is defined by a 3D rotation using a unit vector that specifies the axis of rotation (i.e., the direction of the rotation's normal vector) and a scalar representing the magnitude of the rotation around that axis. While this method is gimbal lock-free, it still suffers from discontinuities (Zhou *et al.*, 2019).

Quaternions representation

Quaternions are 4D vectors that represent 3D rotation without singularities using a normalized 4D vector. Their main idea is defining a rotation around an axis with a unit vector (x,y y, z) by angle θ (Kuipers, 1999):

$$q = (q_1, q_2, q_3, q_3) = (\cos\frac{\theta}{2}, x\sin\frac{\theta}{2}, y\sin\frac{\theta}{2}, z\sin\frac{\theta}{2}) \tag{1.5}$$

Unit quaternions allow smooth interpolation, are numerically stable and are gimbal-lock free (Kuipers, 1999; Grassia, 1998) but suffer from discontinuity (Zhou *et al.*, 2019; Grassia, 1998) and sign ambiguity since q and $-q$ represent the same rotation posing the problem of very different values for the same pose.

6D pose representation

Recent research explores implicit orientation representations to avoid the mentioned pitfalls in the previous representations as they don't allow a proper optimization of the neural network: discontinuities and gimbal locks make the model prone to discontinuous and undefined predictions. For instance, neural networks learn rotation representations that are continuous, differentiable, and unconstrained.

Zhou *et al.* (2019) presented an approach where they replace rotation matrices with an easy to enforce continuous representation. Their idea enforced applying the GS process on the representation itself. From a deep learning perspective, it is equivalent to applying a transformation to the ground-truth value.
The mapping from $SO(3) \subset \mathbb{R}^{3\times 3}$ to the defined 6D representation in $\mathbb{R}^{3\times 2}$ is defined as the function g_{GS} (Zhou *et al.*, 2019):

$$g_{\text{GS}}\left(\begin{bmatrix} | & | & | \\ \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \\ | & | & | \end{bmatrix}\right) = \begin{bmatrix} | & | \\ \mathbf{r}_1 & \mathbf{r}_2 \\ | & | \end{bmatrix} \tag{1.6}$$

The inverse operation is f_{GS} (Zhou *et al.*, 2019):

$$f_{\text{GS}}\left(\begin{bmatrix} | & | \\ \mathbf{r}_1 & \mathbf{r}_2 \\ | & | \end{bmatrix}\right) = \begin{bmatrix} | & | & | \\ \mathbf{r}'_1 & \mathbf{r}'_2 & \mathbf{r}'_3 \\ | & | & | \end{bmatrix} \tag{1.7}$$

where each column $\mathbf{r}'_i$ is given by:

$$\mathbf{r}'_i = \begin{cases} \dfrac{\mathbf{r}_1}{\|\mathbf{r}_1\|} & \text{if } i = 1 \\ \dfrac{\mathbf{r}_2 - (\mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{r}'_1}{\|\mathbf{r}_2 - (\mathbf{r}_1 \cdot \mathbf{r}_2)\mathbf{r}'_1\|} & \text{if } i = 2 \\ \mathbf{r}'_1 \times \mathbf{r}'_2 & \text{if } i = 3 \end{cases} \tag{1.8}$$

With this transformation f_{GS}, the obtained result satisfies the orthogonality constraint and the determinant of the obtained matrix is equal to 1. As a result the obtained matrix $\mathbf{r}'$ can be transformed to three unique angles corresponding to a defined Euler sequence.

In their paper entitled "6D Rotation Representation For Unconstrained Head Pose Estimation" (6DRepNet), Hempel *et al.* (2022) applied this transformation for the head pose transformation task. In fact, they used the following datasets: 300W-LP (Sagonas, Tzimiropoulos, Zafeiriou & Pantic, 2013), ALFW2000 (Huang, Mattar, Berg & Learned-Miller, 2008) and BIWI (Fanelli, Dantone, Gall, Fossati & Van Gool, 2013) to train and evaluate a pose estimation model. These datasets are annotated with a sequence of Euler angles that are transformed according to their defined Euler sequence to a 3 × 3 rotation matrix then to the $6D$ representation following the approach defined by Zhou *et al.* (2019). Their model is composed of a RepVGG backbone (Ding *et al.*, 2021) followed by a linear module that maps the stacked averaged feature maps to a $1D$ vector of 6 values. That vector is then transformed to a 3 × 3 rotation matrix. To optimize the network, they used the geodesic cost function as the train objective instead of the L_2 norm and applied it to the ground-truth and predicted rotation matrices.

The geodesic loss function is defined as:

$$L_G = \cos^{-1}\left(\frac{\mathrm{tr}(R_{pred}R_{GT}^{\top}) - 1}{2}\right) \tag{1.9}$$

with R_{pred} and R_{GT} respectively the predicted and ground-truth 3 × 3 rotation matrices.
While some precedent approaches focused more on landmark-based approaches, Hempel *et al.*

(2022) stated that an accurate head pose estimation will be reliant on the performance of the landmarks positions. The previous landmark-based approaches like HopeNet (Doosti, Naha, Mirbagheri & Crandall, 2020), WHENet (Zhou & Gregson, 2020), FSA-Net (Yang, Chen, Lin & Chuang, 2019) rely on methods that mainly use the Euler angles and quaternions representations. Their approach was the first to use the $6D$ representation for the head pose estimation and showed better performance in terms of Mean Absolute Error (MAE) of the three Euler angles. The model trained on AFLW2000 dataset was tested on the test set of the same database and on the BIWI dataset. The average error of the test MAEs (yaw, pitch and roll angles) was compared with other pipelines trained and tested on the same sets. The Table 1.1 shows the results of the test angle errors (yaw, pitch, roll and their average) for the 6DRepNet model trained on 70% of the BIWI dataset and tested on the remaining 30% compared to the results of previous SOTA methods HopeNet (Doosti *et al.*, 2020), FSA-Net (Yang *et al.*, 2019), TriNet (Cao, Chu, Liu & Chen, 2021) and FDN Zhang, Wang, Liu & Yuan (2020).

Table 1.1 Comparison of head pose estimation methods test errors (°)
Adapted from Hempel *et al.* (2022)

Model	**Yaw**	**Pitch**	**Roll**	**MAE**
HopeNet	3.29	3.39	3.00	3.23
FSA-Net	2.89	4.29	3.60	3.60
TriNet	2.93	3.04	2.44	2.80
FDN	3.00	3.98	2.88	3.29
6DRepNet	**2.69**	**2.92**	**2.36**	**2.66**

These results can be a motivation to develop a similar framework for the separate estimation of both the tibial and the femoral anatomical frames using the $6D$ representation, then inferring the anatomical angles by the Euler sequence from both frames, specifically the flexion angle.

1.1.3 Joint angle-based pose estimation

While most deep learning research works focused on estimating joint coordinates and improving their precision (Paragraph 1.1.1), only a few studies that are presented in this paragraph used deep learning to directly infer joint angles. Kidzinski *et al.* (2020) built their own dataset consisting of 1792 videos of 1026 unique patients diagnosed with cerebral palsy for a clinical gait analysis processed by Gillette Specialty Healthcare. Rather than using the traditional clinical workflow—which involves a dedicated motion capture setup and clinicians placing reflective markers on the body—they opted for a more accessible approach using a single-view camera to record the trials. The figure 1.2 from their paper briefly explains the general workflow:

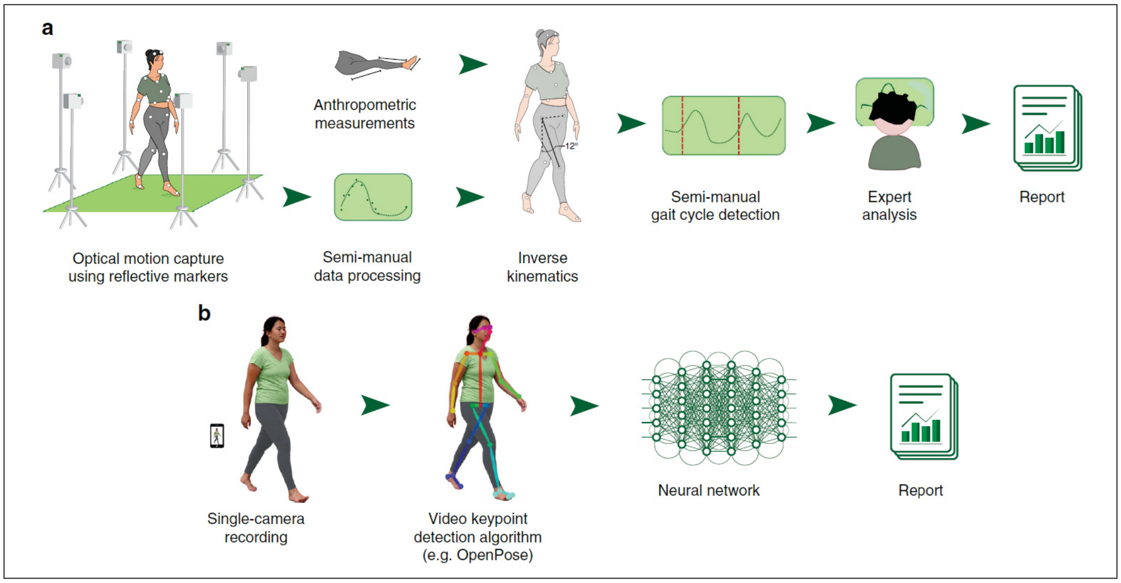

a
Anthropometric measurements
12°
Optical motion capture using reflective markers
Semi-manual data processing
Inverse kinematics
Semi-manual gait cycle detection
Expert analysis
Report
b
Single-camera recording
Video keypoint detection algorithm (e.g. OpenPose)
Neural network
Report

Figure 1.2 Clinical workflow (a) and video-based workflow (b) comparison
Taken from Kidzinski *et al.* (2020)

The clinical workflow (a) is used to generate the training data (image, marker-based data and gait parameters) and the reference test data while the video-based workflow (b) shows the inference pipeline to predict the parameters from the OpenPose keypoint's signals.

For each gait parameter, there is a model that is based on a series of convolutional blocks followed by dense layers to predict that final gait parameter. The input is time series of the pose estimates

and the prediction is the gait parameter of interest. To generate ground-truth data for training, each trial was also recorded with a traditional optical motion capture system. Reflective marker data from these recordings were used to compute biomechanical metrics such as the knee flexion angle at maximum extension, following established practices (Kadaba, Ramakrishnan & Wootten, 1990). Since the target parameters are trial-specific rather than image-specific, the authors assumed that the trial-to-trial variability is negligible. Under this assumption, comparing the model's predictions—based on OpenPose outputs—with the ground-truth motion capture measurements provides a valid basis for evaluating model performance.
Their best model showed a MAE of 4.8° for the maximum extension knee flexion angle and a correlation of 0.83 with the angle obtained from the ground-truth motion-capture data compared to a 0.51 correlation for the angle formed by the lower limbs segments of the projected 2D joints.

Inspired by this idea of using the joints time series to predict gait parameters, Le & Pham (2024) transforms the input OpenPose time series predictions into joint embeddings which are fed to a transformer-based architecture composed of spatial attention and temporal attention blocks. The spatial attention component of this architecture allows it to gather joint dependencies information unlike the CNN architecture by Kidzinski *et al.* (2020) who only focuses on temporal dependency (between frames).
The experiments were carried on the same dataset following the same procedure. The results showed an augmentation of 1.04% of the knee flexion angle correlation with the ground-truth values and a reduction by 1.54% of its MAE.

Similarly, Dinh *et al.* (2025) introduced a CNN-based architecture known as the Dual Pattern Gap (DPG) model. This approach utilizes two preprocessed image-based datasets derived from skeletal data generated by OpenPose (joint predictions). The first dataset represents all 25 body landmarks along with 8 side parameters, while the second focuses on just 3 lower-body landmarks: the hip, knee, and ankle. Both datasets are simultaneously input into a series of convolutional blocks. The resulting feature maps are then concatenated into a single feature vector, which is passed through a linear classification module to predict individual gait parameters. The experimental results on the same Gillette Specialty Healthcare dataset showed an improvement of the MAE by 12% compared to (Le & Pham, 2024) and by 13.4% compared to (Kidzinski *et al.*, 2020) for the Knee flexion angle.

Stenum *et al.* (2021) made a study to assess the reliability of deep learning models, specifically OpenPose in identifying some gait parameters. One of the parameters is the right and left knee flexion angles. The dataset was collected using Vicon motion capture cameras and four video cameras for the right and left sagittal planes, frontal plane and back plane. Using the reflexive markers positions, the coordinates of the joints are computed and based on those coordinates the flexion angle formed by the lower limbs segments is extracted.
For the same sequence, they applied OpenPose on both left and right view cameras to predict the keypoint coordinates and then compute the flexion angle from those coordinates using the same method applied in the case of motion capture data.

The table 1.2 shows their obtained error analysis and the correlation between the three measurements: Motion Capture system (MC), OpenPose-based measurement from the left camera (C_L) and OpenPose-based measurement from the right camera (C_R). All statistics were made for both left and right knees for 31 participants (31 unique trials).

Table 1.2 Knee angle statistics
Adapted from Stenum *et al.* (2021)

Metric	**Side**	**MC–C_L**	**MC–C_R**	C_L–C_R
MAE (°)	Left	5.1 ± 2.1	5.5 ± 2.4	3.5 ± 1.1
	Right	5.6 ± 2.7	5.6 ± 2.9	3.8 ± 1.4
Correlation (r)	Left	0.984 ± 0.012	0.983 ± 0.012	0.992 ± 0.005
	Right	0.980 ± 0.007	0.979 ± 0.027	0.989 ± 0.008

Contrary to these approaches, Jamsrandorj *et al.* (2022) proposed the idea of estimating the knee and elbow joint angles directly from a RGB image without any joint-based coordinate estimation.

First, they built a dataset of 15 participants performing four daily actions: squat, walking in place, one leg stand for both legs and arms bending parallel to the ground. The experiment was

carried with a motion capture setup consisting of synchronized six optical motion cameras and four RGB cameras (four different views). Using this setup, the ground-truth knee and elbow flexion/extension angles were computed as labels for the sequences.
The framework is composed of ResNet18 pretrained on ImageNet backbone to extract the features followed by a LSTM module composed of four bidirectional LSTM layers and then two fully connected layers to predict the four angles. One version of the model was trained on all the actions while the other version trains a different model instance for each action.

For the walking action, the best action-independent model achieved a MAE of 5.15° and 7.88° respectively on the right and left knee flexion angles and a Pearson's correlation coefficient (PCC) of 0.933 for both knee flexion angles. In their work they also demonstrate the model capacity of producing similar results when trained with data from only a specific camera or on instances of all the actions combined.

1.2 Deep learning techniques

In this Section 1.2, we review key deep learning fundamentals that are potentially useful in computer vision problems and more specifically in this case study.

1.2.1 Image processing backbones

In human pose estimation and in computer vision in general, many pretrained architectures were proposed in the literature to optimize image feature extraction from large-scale datasets.

ResNet

Residual Networks (ResNet) (He *et al.*, 2016) is a groundbreaking backbone in the field of deep learning because it allowed training exceptionally deep neural networks at the time without degrading the performance. ResNet's main contribution is the residual learning framework, implemented through the skip connection units. Instead of stacking multiple layers sequentially to learn a mapping H(x) directly, ResNet proposes a residual mapping to those layers denoted as:

$$F(x) = H(x) - x \tag{1.10}$$

where $H(x)$ is the output of the residual block formed by the Identity Shortcut.

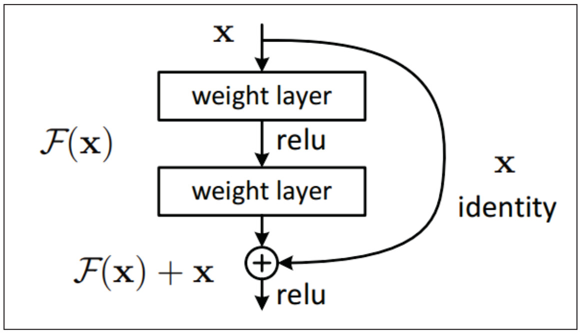


Figure 1.3 Residual bloc of ResNet
Taken from He *et al.* (2016)

The original input x is then added back to F(x) at the output of the block, so the effective output is F(x) + x.

The ResNet architecture is composed of mainly two types of residual blocs: Basic blocs and bottleneck blocs. The basic block is a concatenation of 2 convolutional operations of kernel size 3 × 3. The bottleneck is composed a 1 × 1 convolution to reduce dimensionality, a 3 × 3 convolution to perform feature extraction on reduced dimensionality and a 1 × 1 convolution to restore the dimensionality. There are many variants of ResNet depending on the number of blocks used. Resnet50 is one of the the most widely used architectures and is composed of:

- Stage 0: Takes an input image of size 3 × 224 × 224 and is based on a 7 × 7 convolution and a 3 × 3 max pooling layer. The output shape of this stage is 64 × 56 × 56.
- Stage 1: 3 bottleneck blocs. The output shape of this stage is 256 × 56 × 56
- Stage 2: 4 bottleneck blocs. The output shape of this stage is 512 × 28 × 28
- Stage 3: 6 bottleneck blocs. The output shape of this stage is 1024 × 14 × 14
- Stage 4: Consists of 3 Bottleneck Blocks. The output shape of this stage is 2048 × 7 × 7
- Stage 4: A final average pooling layer to output a stacked feature vector of size 2048.

ResNet is commonly used with weights pretrained on the ImageNet dataset (Russakovsky *et al.*, 2015), allowing the model to have a powerful feature extractor. This method, called transfer learning, is very useful, especially in cases with limited data. It makes the model convergence faster and improves the performance.

RepVGG

RepVGG (Ding *et al.*, 2021) is a novel convolutional neural network architecture that combines the fast-inference and hardware efficiency of VGG-style networks with the high accuracy provided by multi-branch architectures like ResNet. Its core innovation lies in structural re-parameterization.

In training phase, RepVGG has a similar structure to ResNet by using 3 branches: the main 3×3 convolution, identity and an extra 1×1 branche. During inference, it follows the VGG structure by using a single path topology reducing the inference time.

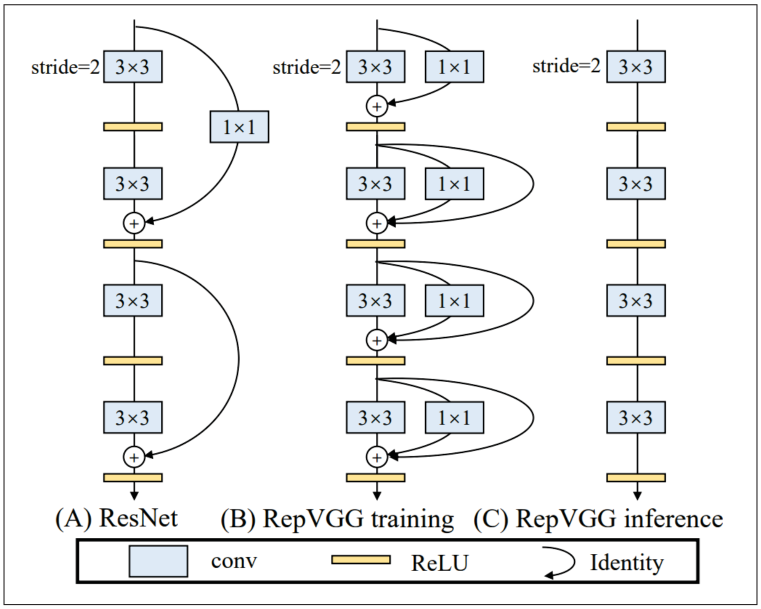


Figure 1.4 The first 4 layers of a RepVGG stage
Taken from Ding *et al.* (2021)

The complex multi-branch structure of RepVGG helps boost the representational capacity of the model for training. The output of the three branches at each stage are followed by batch

normalization layers to get into a single an equivalent 3×3 convolutional layer.
The RepVGG-B1G2 architecture is equivalent to ResNet-50: it has the same stages output shapes and a faster inference speed.

1.2.2 Feature maps and constraints

Features maps are the outputs of a convolutional layer after applying a filter to the input data. It translates to the spatial arrangement of the learned features mostly the edges, textures, etc (Goodfellow, Bengio & Courville, 2016). Since features maps contain crucial information extracted from images at intermediate stages of the backbones like ResNet (He *et al.*, 2016) and RepVGG (Ding *et al.*, 2021), extracting them and ensuring they converge in the right direction can help the model learn in a more effective and optimized way.

Multiple methods used the intermediate supervision of certain features or labels at some stages of a trained network. Lee, Xie, Gallagher, Zhang & Tu (2015) added auxiliary Support Vector Machine (SVM) supervised classifier heads at intermediate layers to a standard CNN classifier. The final targets were used to help the optimization and improve feature learning. It helped solve the vanishing gradients issue and guide feature extraction early at the network. Similarly, GoogLeNet (Szegedy *et al.*, 2015) used an inception network with intermediate classification after two inception modules to help regularize training and improve convergence.
In the context of HPE, HRNet (Sun *et al.*, 2019a) applied supervision to heatmaps at intermediate resolutions in addition to the final heatmaps supervision, helping it maintain spatial precision across branches. Newell *et al.* (2016) also applied intermediate supervision by progressively refining the predictions after each Hourglass module and supervising a predicted joint heatmap. Some other methods used intermediate supervision information on pseudo-targets like Zagoruyko & Komodakis (2017) did with a semi-supervised method where he used the supervision of intermediate features in the student network maps to match the attention maps from a teacher network.

1.2.3 Data augmentation

Data augmentation is an important technique in deep learning to help the model generalize on more diverse data and help reduce overfitting during training (Shorten & Khoshgoftaar, 2019a). Shorten & Khoshgoftaar (2019b) made a survey about image data augmentation for deep learning models. The authors presented a taxonomy of the methods used in literature and classified then into 2 categories: Basic approaches and advanced approaches with each one classified on its own to subcategories.

The basic approaches primarily include basic image manipulations (such as cropping, flipping, rotating, injecting noise or blur, changing brightness and contrast, etc.), image erasing (deleting some parts of the image) and image mixing (mixing two images with the appropriate labeling transformation).

Advanced approaches include applying augmentations on the feature space or using generative models like GANs

For this study on knee flexion angle from images; advanced approaches, image mixing and image erasing are not appropriate because they are more likely to hurt the data distribution and produce unrealistic data. Kinematic data is a very accurate type of dataset and some transformations will just produce invalid samples that will make the training diverge.

The focus is more towards what pose estimation approaches used to limit the possible combination of augmentations and select what is appropriate for this project. The image data augmentations of some of the most used HPE methods for the datasets COCO (Lin *et al.*, 2014), MPII (Andriluka, Pishchulin, Gehler & Schiele, 2014), Human3.6M (Ionescu *et al.*, 2014), HumanEva-I (Sigal, Balan & Black, 2010), FLIC (Sapp & Taskar, 2013) and MPI-INF-3DHP (Mehta *et al.*, 2017) are summarized in the Table 1.3:

The rotation angle range and the zoom scales varies according to the model and the dataset. The flip transformation is helpful in most of these pipelines because the dataset has images of the subjects from different perspectives and camera angles. In our case it is different since the trials

Table 1.3 Overview of the image data augmentations

Pipeline	**Rotation**	**Zoom**	**Flip**	**Datasets**
SimpleBaseline	✓	✓	✓	COCO, MPII
VideoPose3D	✗	✗	✓	Human3.6M, HumanEva-I
HRNet	✓	✓	✓	COCO, MPII
StackedHourglass	✓	✓	✗	FLIC, MPII
Weakly-supervised Approach	✓	✓	✗	MPI-INF-3DHP
DeepPose	✗	✓	✓	FLIC, LSP

are recorded uni-directionally: the person is standing parallel to the camera and looking towards the right side of the camera frame.

1.3 Model validation and performance evaluation methods

Selecting a model architecture alone is not sufficient. It is fundamental to have a well-defined validation methodology to justify the model selection depending on various factors like the hyperparameter choice and the data partitioning. This ensure that the methodology effectively learns relevant patterns from a training set and is reliably evaluated on a separate test set. Moreover, defining appropriate evaluation metrics is also important to establish a clear objective and a consistent comparison basis. In this Section 1.3, we explore a data-driven validation procedure for the model (Subsection 1.3.1) and discuss potential evaluation metrics (Subsection 1.3.2) to be used in this study.

1.3.1 Cross-validation (CV)

CV is a crucial technique that systematically splits data for training and validation to assess the generalization (Kohavi, 1995). In medical field and healthcare, datasets are likely to face challenges like class imbalance, heterogeneity, limited number and data leakage (Litjens *et al.*, 2017; Kapoor & Narayanan, 2023). In this subsection, we show how CV bypasses these challenges mostly encountered by other traditional data splitting techniques.

Bradshaw *et al.* (2023) wrote a comprehensive article on the different validation techniques of AI algorithms including CV, their uses cases, advantages and limitations.

A typical and simple way of training and testing a model is a separation in train and test sets. Xiao *et al.* (2018); Pavllo, Feichtenhofer, Grangier & Auli (2019) split the 7 subjects Human3.6m dataset (Ionescu *et al.*, 2014) in train and validation and considered the best validation score checkpoint as the final model and the test performance is set to that score. Similarly, Wang *et al.* (2024) used the same split for the Human3.6M dataset (Ionescu *et al.*, 2014) and split the CMU Mocap dataset (C. G. Lab, 2003) into respectively 133 and 11 subjects for training and validation. With this split, there is no possibility of choosing the optimal model trained on the train set since its performance would be biased towards the train data (Bradshaw *et al.*, 2023). This method is called One-time train-test split.

To resolve this issue, another split with an extra set is introduced called the one-time train-val-test split where the validation set is used to select the trained model with the optimal performance for the final testing on the test set (Bradshaw *et al.*, 2023). The limitation of the train-validation-test split is the fact that the validation set might bias the results if it is not well representative of the dataset (Bradshaw *et al.*, 2023). Hammerla, Halloran & Plötz (2016) use three human recognition databases with a low number of subjects (4, 9 and 10) and split them each separately into a train, validation and test set with unique trials for each one. Similarly, Hannun *et al.* (2019) and Rajpurkar *et al.* (2017) split respectively a 29k patients ECG records dataset and a 30k patients Chest X-ray dataset with no overlaps.

The principle of CV (Pedregosa *et al.*, 2012) is commonly used to evaluate a model's performance across the entire dataset, rather than relying on a single split. The standard approach involves dividing the dataset into K folds and computing the average validation score across all folds. This average serves as a reliable indicator of the model's overall performance. The hyperparameter configuration that holds the best average score is then selected as the optimal one. CV can be used for three purposes: estimating a model's performance by averaging results over *n* validation folds, selecting the best model (for example, choosing between SVM and Random Forest based on performance over n folds), and tuning the hyperparameters of a defined model (Refaeilzadeh, Tang & Liu, 2009).

It is important to distinguish between Train/Test CV (test-folded CV) and Train/Validation CV with holdout test. The former evaluates different models trained on various training subsets and tested on different test folds—meaning the average performance may reflect models with potentially different hyperparameters. In this method, the reported performance is not that of a single model but of the entire modeling pipeline. The latter performs CV on different validation folds to determine the best model architecture suited for the entire available dataset, based on the average validation scores; then the final model is trained according to the chosen architecture and hyperparameter configuration on the whole training set and tested on the holdout set. In this case, there is a unique performance value for a single model on a single test set.

For the test-folded CV, Lombardi *et al.* (2021) used a MRI Scan dataset collected from 17 different sites in their model and used the LeaveOneOut (LOO) (Pedregosa *et al.*, 2012) train/test CV strategy by assigning one site data to each fold. NN-UNet (Isensee, Jaeger, Kohl, Petersen & Maier-Hein, 2021) is a neural network trained and evaluated on MRI and CT scans. The approach uses a five-fold train/test CV.

In nested CV (Wainer & Cawley, 2021), the dataset is first divided into n folds in what is known as the outer loop. Each fold serves, in turn, as a test set (Test_i), while the remaining folds form the corresponding training set (Train_i). For each set Train_i, an inner CV is performed independently to select the best model and optimize its hyperparameters, treating Train_i as if it were the entire dataset for the inner CV. Once the optimal model and its hyperparameters are identified, it is retrained on the entire Train_i and evaluated on the associated Test_i. Consequently, each outer fold yields a different model instead of one single final model. For example, a nested CV with an outer loop of 3 folds random forest may result in different hyperparameter values across folds, such as $n\_estimators = 5$ for $i = 1$, $n\_estimators = 10$ for $i = 2$, and $n\_estimators = 20$ for $i = 3$.

In contrast, traditional k-fold train/test CV (no holdout test) can lead to overfitting, as the same data are used both for model tuning and for performance evaluation. This overlap may introduce data leakage, resulting in an overly optimistic estimation of model performance. The severity of this issue depends primarily on the dataset size and the stability of the model (Cawley & Talbot, 2010). Nested CV avoids this problem by keeping the test data entirely separate from the model selection process.

One approach that adopted this strategy is (Lombardi *et al.*, 2022). The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset (Petersen *et al.*, 2010) was used, which contains MRI scans from 57 different sites. The authors employ a Leave-One-Out Train/Test CV at the outer CV test loop, where one site is used for testing in each iteration. For training, they perform hyperparameter fine-tuning on the remaining 56 sites using an internal Train/Validation CV with $K = 3$.

To summarize these approaches, the Figure 1.5 reproduced from (Bradshaw *et al.*, 2023) help understand the high-level of these different methods.

In order to increase the number of validation iterations to select the best model in a test-holdout scheme, a simple solution would be to increase the number of folds K.

Another approach would be to reshuffle the CV distribution for n number of iterations. In other terms, the train/validation CV is done n times where we reshuffle the folds each time. This method is called Repeated CV (GSS-CV) (Pedregosa *et al.*, 2012). The Figure 1.6 shows a comprehensive scheme of the method.

Krstajic, Buturovic, Leahy & Thomas (2014) emphasized on the importance of using repeated CV in model and hyperparameter selection or nested CV in model assessment. In fact, they proved that repeating CV improves the reliability of the model and increases the confidence of its predictions. The tests of the study were carried on seven QSAR datasets (Kuhn, 2012) using multiple algorithms.

With repeated k-fold the noise from the model's estimates is reduced. The more combinations (repetitions) are used, the more precise the model fine-tuning becomes, as it reduces the variance between validation scores (Bradshaw *et al.*, 2023).

De Filippi *et al.* (2021) used a $n = 10$ repeated 8-fold CV on their own acquired EEG data

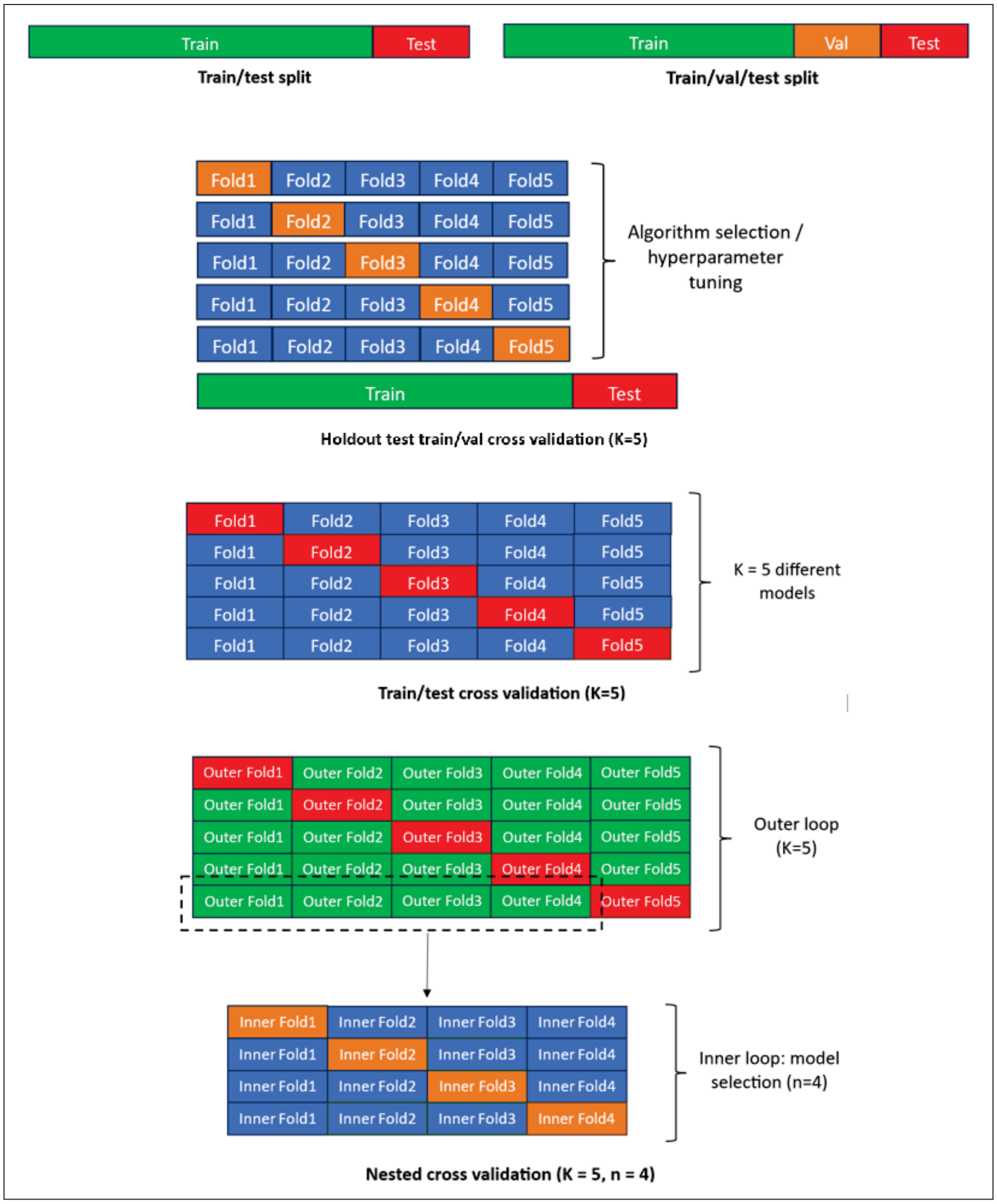


Figure 1.5 Traditional validation and CV approaches overview
Adapted from Bradshaw *et al.* (2023)

then averaged the model performance on all the 80 runs for model/hyperparameter selection. White & Power (2023) also applied a $n = 5$ repeated 6-fold CV on two EEG public datasets and reported the average performance on the 30 runs.

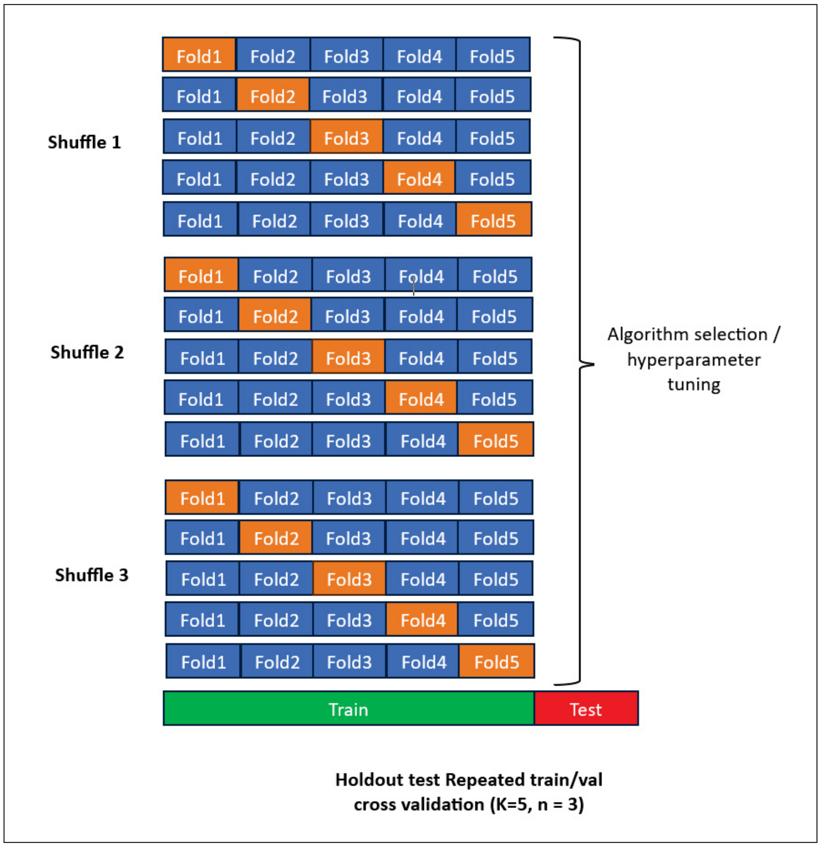

Shuffle 1
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Shuffle 2
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Shuffle 3
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Fold1 Fold2 Fold3 Fold4 Fold5
Algorithm selection / hyperparameter tuning
Train
Test
Holdout test Repeated train/val cross validation (K=5, n = 3)

Figure 1.6 Repeated K-Fold CV with holdout test set

It is important to note that in these studied approaches in literature, datasets with group-specific characteristics are folded based on that feature. For example, datasets based on trials must be split or folded without any data overlap between patients/participants: each patient/participant can only belong to one fold/set. This way of initiating and randomizing data splits/folds is called group-based split. More specifically, Pedregosa *et al.* (2012) call this Group K-fold CV. This definition also extends to all the variants of CVs notably repeated Group K-fold CV.

1.3.2 Evaluation metrics

After obtaining a method's predicted angles, it is essential to define the evaluation metrics used to assess the accuracy and performance of these predictions.

To evaluate kinematic data and in particular angular data, most approaches use the MAE and

Pearson correlation coefficient (PCC) (Jamsrandorj *et al.*, 2022; Stenum *et al.*, 2021; Dinh *et al.*, 2025; Kidzinski *et al.*, 2020; Le & Pham, 2024) to evaluate the performance of the pipeline.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{\theta}_i - \theta_i\right| \tag{1.11}$$

$$PCC = \frac{\sum_{i=1}^{n}\left(\hat{\theta}_i - \bar{\hat{\theta}}\right)\left(\theta_i - \bar{\theta}\right)}{\sqrt{\sum_{i=1}^{n}\left(\hat{\theta}_i - \bar{\hat{\theta}}\right)^2}\sqrt{\sum_{i=1}^{n}\left(\theta_i - \bar{\theta}\right)^2}} \tag{1.12}$$

In (Hempel *et al.*, 2022), the authors train their model on the geodesic loss and then infer the Euler angles from the final rotation matrix with the three angles possibly between -180° and 180° depending on the Euler sequence. To avoid high trivial errors between extreme angles, a modified version that takes into account the periodicity of the angles was used. For each one of the three Euler angles:

$$\mathrm{MAE} = \min\left(\left|\hat{\theta} - \theta\right|, \left|\hat{\theta} + 360^\circ - \theta\right|, \left|\hat{\theta} - 360^\circ - \theta\right|, \left|\hat{\theta} + 180^\circ - \theta\right|, \left|\hat{\theta} - 180^\circ - \theta\right|\right) \tag{1.13}$$

1.3.3 Statistical mapping

Assessing results solely on descriptive statistics such as the mean and standard deviation might not always give full insights on the obtained results. Statistical testing tools help to evaluate the statistical significance of certain assumptions/hypothesis rigorously. In this Subsection 1.3.3, we introduce a statical testing method (inferential statistics) appropriate for gait data.

Statistical mapping is a statistical tool for analyzing spatially or temporally extended data in fields like neuroimaging and biomechanics. Instead of using discrete summary statistics (overall mean for example), this technique provides a continuous statistical inference across the entire data domain for a more comprehensive interpretation and to identify the statistical significance of the regions and periods (Pataky, 2010).

The 1D-SPM (SPM1D) (Pataky, 2011) is used to analyze one-dimensional time-series data, in particular biomechanics to evaluate statistical significance or test certain hypotheses where data consists of continuous time series curves. It uses random field theory to account for temporal smoothness and correlation. SPM1D allows for the statistical testing across the entire time series.

Let's suppose we have N-1 dimensional signals S_1, S_2, ..., S_N of length T each. Applying SPM1D-t (SPM1D based on a t-test) accounts for applying t-test on all the samples $S_i(k)$ at each instant $k \in [1, N]$. The obtained result is a time series curve of the statistical test value of length T.

1.4 Database processing techniques

Since the objective of the study is to estimate certain kinematic parameters (angles) from a monocular source in the context gait analysis, the target data primarily consists of the angles to be predicted. Each trial (acquisition) represents a time series of captured images accompanied by corresponding annotations of the kinematic parameters for each frame. On one hand, it is essential to address the presence of markers in the images because they may introduce bias (Section 0.2) in the estimation method. On the other hand, standardized data processing techniques commonly used in gait analysis are applied for more reliable and consistent results. First, we start by reviewing image inpainting techniques (Subsection 1.4.1) to remove the KneeKG out of the images. Then, we study the normalization of gait cycles (Subsection 1.4.2), a technique widely used in biomechanics analysis.

1.4.1 Image inpainting

Image and video inpainting involves reconstructing missing or occluded areas in digital images or videos by filling them with realistic and coherent content (Quan, Chen, Liu, Yan & Wonka, 2024). In this project, inpainting is an important tool to create an alternate dataset consisting of images/videos without the KneeKG tool and accessories as they might cause distortions during training (example shown on figure 1.7). The input image of the inpaint module is called "damaged image" and the output is called "restored image".

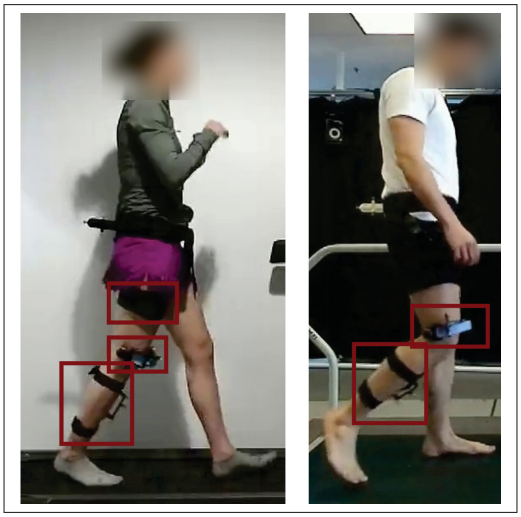


Figure 1.7   Example of KneeKG accessories shown on images

Before deep learning breakthrough in 2014, inpainting techniques were based on traditional computer vision techniques and focused more on repairing minor fixes like scratches and noise from the image but they were unable to process complex corrections and changes. Deep learning-based allowed the production of more accurate and visually plausible results compared to traditional methods. These methods leverage CNNs and GANs to understand complex image structures, textures, and colors and fill larger and more complex missing regions (Salem, 2021; Quan *et al.*, 2024).

The general idea behind deep learning-based image inpainting is to map the input image through an encoder to obtain a compressed latent space for the decoder to reconstruct the restored image (Xu *et al.*, 2023). We optimally want an automated process that removes the object of all the video (or images of the same sequence).

The early image inpainting works based on deep learning focused on convolutional architectures. Cai *et al.* (2017) introduced fully convolutional neural network trained on pairs of ground truth (GT) and masked images to detect and fill missing regions. Although the model is effective for inpainting certain missing areas after training, it lacks robustness as it is only limited to specific missing region patterns and does not generalize to complex and random shapes. Sun *et al.*

(2019b) designed a two-stage framework: a content-aware module to predict the global content of the missing part and a texture module to iteratively upscale the texture of the inpainted region.

In general, CNN-based approaches generated blurry results mainly due to the nature of the euclidean distance used in optimizing the inpainting loss (Sun *et al.*, 2019b). With the introduction of Generative Adversarial Networks (GANs), the added adversarial loss allows to add a complementary loss component that helps generating realistic results due to the capacity of the generator in considering global information and performing feature extraction for the image generation.

One of the first advanced GAN architectures in which the inpainting efficiency was tested and confirmed is PatchGAN (Isola, Zhu, Zhou & Efros, 2017; Demir & Ünal, 2018) whose discriminator runs on multiple patches of the same image then averages the outputs of the same patches to output a final classification as real or fake. Zeng, Fu, Chao & Guo (2019) used a pyramid-context encoder (PEN-NET) based on the U-NET structure with an attention module to encode the contextual semantics and a multi-scale decoder. Other approaches (Salem, Mahdi & Abbas, 2019; Yu *et al.*, 2019) refined the GAN architectures by improving the semantic consistency, texture and color of the generated images.

Suvorov *et al.* (2022) introduced a GAN-based resolution-robust inpainting network named "LaMa" that replaces conventional convolutions with Fourier Convolutions (FFCs), which are better at capturing global context and maintain high performance across varying resolutions even bigger than the ones the model was trained on. The model generalizes well on different objects in different contexts without the need for further fine-tuning; which is suitable for the KneeKG case since it is an uncommon object. What makes LaMa the reference SOTA model in image inpainting despite some later methods having a slightly better performance is its fast inference time and lightweight characteristic despite his effective performance mainly due to being a single-stage network.

The generator described in the Figure 1.8 takes as input the concatenation of a restored (masked) image and the mask itself then outputs the inpainted image. This module is composed of 3 main blocs: a fully Convolutional downscaling block, a nine stacked Fast Fourier Convolutions-based residual block and a fully Convolutional upscaling bloc mirroring the downscaling part.

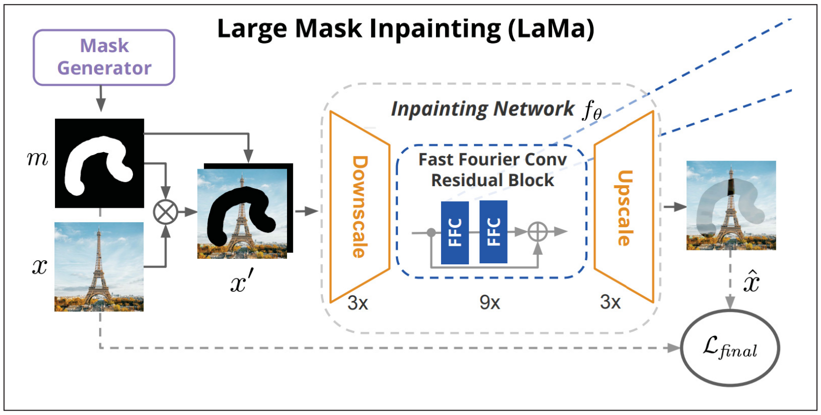

Large Mask Inpainting (LaMa)
Mask Generator
m
x
x'
Inpainting Network f_θ
Downscale
Fast Fourier Conv Residual Block
FFC
FFC
Upscale
3x
9x
3x
$\hat{x}$
$\mathcal{L}_{final}$

Figure 1.8 LaMa generator architecture
Taken from Suvorov *et al.* (2022)

The FFC module detailed in the Figure 1.9 below is based on Fourier transformations to extract global features and local features with the standard Convolutional layers.

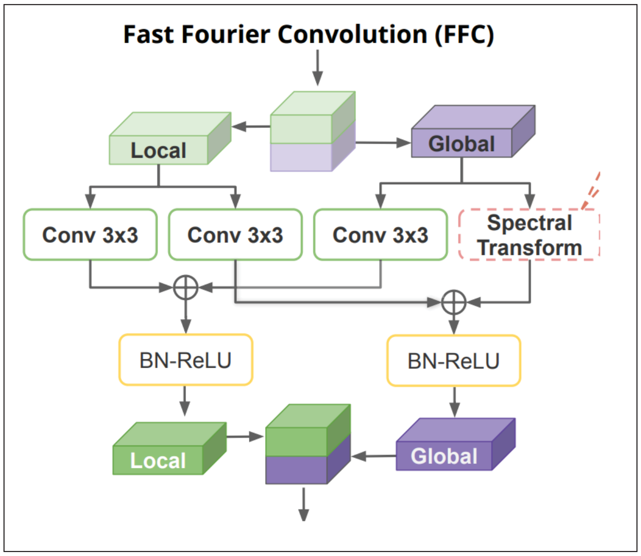


Figure 1.9 FFC residual bloc
Taken from Suvorov *et al.* (2022)

To train this inpainting model, a well-designed loss function that guides the generator effectively is required. According to Suvorov *et al.* (2022) simple pixel-wise comparisons are insufficient due to the diversity of plausible pixel completions for the same missing areas. The loss must account for both global image coherence and local detail accuracy to ensure realistic and contextually appropriate inpainting results. The loss function used to optimize the model is composed of 3 different terms;

- High receptive field perceptual loss $\mathcal{L}_{\text{HRFPL}}$: A novel adaptation of perceptual loss (Johnson, Alahi & Fei-Fei, 2016) utilizing a pretrained network with a large receptive field, designed exclusively to focus on high-level patterns and global structural features: optimizes the high level details. It is composed of the generator and discriminator losses.
- Adversarial loss $\mathcal{L}_{\text{ADV}}$: The adversarial learning-based loss responsible for generating real-looking inpainted pixels: optimizes the low-level details.
- Discriminator-based perceptual loss $\mathcal{L}_{\text{DiscPL}}$: stabilizes training by optimizing the features of the discriminator network.
- gradient penalty R_1: a regularization term to stabilize the adversarial structure of the network

The final loss is a weighted sum of the mentioned terms:

$$\mathcal{L}_{\text{final}} = \kappa\mathcal{L}_{\text{ADV}} + \alpha\mathcal{L}_{\text{HRFPL}} + \beta\mathcal{L}_{\text{DiscPL}} + \gamma R_1 \tag{1.14}$$

The model Big Lama (the main pretrained model) is publicly available on the paper Github page.

1.4.2 Gait cycles

The gait cycle is the fundamental unit of human motion that defines two consecutive same events of the same lower limb during walking activity. The start of the gait cycle is commonly defined by the initial contact called heel strike of one foot and it is marked by the next initial contact of the same foot (Perry & Burnfield, 2010).

The gait cycle is divided into two main phases:

- The stance phase: it includes the period where the foot is still in contact with the ground and lasts for approximately 60% of the gait cycle.(Perry & Burnfield, 2010). It is subdivided into sub-phases (Kharb *et al.*, 2011):
  - Heel strike: the moment when the foot touches the ground.
  - Loading response: from the initial contact until the other foot is lifted.
  - Mid-stance: the period when the body weight is fully supported by a single leg.
  - Terminal stance: The phase from the rise of the heel and ends just when the other foot is about to make the initial contact.
  - Pre-swing phase: the final period of the stance phase. It starts just after the terminal stance and ends when by the toe-off from the ground.
- The swing phase: the period when the foot is not in contact with the ground. It represents about 60% of the gait cycle. Its sub-phases include:
  - The initial swing: from the end of the stance phase until the maximal knee flexion.
  - Mid-swing: when the swinging leg passes directly beneath the body.
  - Terminal swing: from mid-swing until the next limb initial contact initiating the next gait cycle.

A common practice to analyze the gait cycle effectively is to normalize it by representing it as percentage time series from 0% to 100% to eliminate variability caused by different walking speeds and other factors (Perry & Burnfield, 2010). The cycles are called normalized gait cycles (NGCs).

After computing the NGCs, it is possible to compute the mean gait cycle (average of all gait cycles) for the ground-truth and predicted angles and then the corresponding gait parameters (Perry & Burnfield, 2010; Na & Buchanan, 2019):

- Flexion heel strike angle at 0%: The angle at the start of the cycle (beginning of the stance phase) where the foot is in the first contact with the ground.
- Peak Flexion angle during loading response: refers to the maximum flexion of the knee shortly after the heel strike. It occurs during the loading response phase.

- Peak extension angle in terminal stance: refers to the position where the knee its most extended position
- Peak flexion angle in swing: it is the maximal angle of knee bending that happens during the swing phase.

1.5 Synthesis and Gaps

In this chapter, we examined the current state of research in computer vision-based approaches for biomechanical analysis, with a particular emphasis on estimating knee joint kinematics from monocular image or video data.

The majority of existing studies focus on enhancing the estimation of joint coordinates in 2D or 3D space. However, relatively few works have explored the use of deep learning and pose estimation techniques to directly assess lower-limb kinematics, such as the knee flexion/extension angle. Most approaches rely on a two-step process: first estimating joint coordinates, then computing joint angles based on these predictions. The accuracy and reliability of such methods are inherently dependent on the precision of the joint localization. Additionally, inconsistencies in anatomical angle definitions across clinical practices may introduce variability in model outputs. As a result, the direct estimation of gait parameters—such as the knee flexion/extension angle—from monocular images or video remains a relatively underexplored and promising area of research.

Moreover, there is a lack of public dataset that uses clinically validated systems such as KneeKG for data acquisition, which ensures high-quality ground truth for training and evaluation. Integrating such systems with appropriate deep learning architectures might offer a promising path towards clinically relevant models.

In addition, SOTA HPE approaches that used marker-based pose datasets used a straightforward train/test or train/validation/test split. This is often due to the limited number of participants or patients available in these datasets. However, research in other medical domains has adopted more rigorous and less biased model evaluation strategies, particularly CV techniques, to improve the reliability of performance assessment and model generalization.

Furthermore, image inpainting remains significantly underutilized in the context of lower-limb pose estimation. Current models are typically trained and validated on image or video data containing reflective markers and associated motion capture equipment. Integrating inpainting techniques could enhance robustness by allowing the model to better generalize to less controlled or equipment-free environments.
Gait cycles normalization plays a fundamental normal in creating a universally normalized data for gait analysis independently from the experiment duration.

Based on the highlighted gaps and limitations in the literature, the next chapter 2 introduces the research problem and sets forth the specific objectives of this work.

CHAPTER 2

RESEARCH PROBLEM AND OBJECTIVES

Accurate assessment of knee joint kinematics is critical in clinical diagnostics and biomechanical research (Winter, 2009). Marker-based motion capture systems with optical systems and wearable sensors often require specialized equipment, are costly, and may constrain natural movement (Baker, 2006; Wade *et al.*, 2022). The KneeKG, a marker-based system to collect kinematic data, remains largely limited to clinical environments due to hardware requirements. According to the literature review of Subsection 1, some recent advancements in deep learning and computer vision showcase that it may be possible to estimate joint angles or other gait parameters from a partial skeletal information obtained with joint pose estimation models like OpenPose; which is a low-cost and accessible solution. However, estimating knee kinematics such as the flexion/extension angles from 2D images remains a challenging task due to high variability in human gait, occlusions, and lack of depth information (Antognini *et al.*, 2025). Furthermore, while many deep learning models focus on pose estimation, fewer have addressed direct regression of joint angles with high precision.

In partnership with Emovi and Moveck, this project aims to address the gaps presented in the literature review (Chapter 1) and study the feasibility of transitioning from marker-based to markerless estimation of knee kinematics by developing a deep learning pipeline capable of predicting the the KneeKG extracted flexion angles from single-view RGB images that are within the clinically accepted norms (considered as clinically accurate). By incorporating image inpainting techniques and leveraging the KneeKG system for ground truth labeling, this work seeks to contribute a novel and practical approach to vision-based knee kinematics. The study also emphasizes the importance of a comprehensive validation framework to evaluate model accuracy, generalization across subjects, and interpretability of results.
The specific objectives of this research are:

- To develop a clinically accurate ((McGinley *et al.*, 2009)) deep learning regression model, capable of predicting knee flexion angles from monocular RGB images.

- To design a rigorous validation scheme for model fine-tuning, performance evaluation, and generalization analysis across different subjects.

In the next Chapter 3, we conduct a preliminary study to examine the feasibility of considering the flexion angle as a possible to estimate parameter from a 2D monocular source.

CHAPTER 3

PRELIMINARY STUDY: ANGLE AND LANDMARKS ANALYSIS

Introduction

The motivation behind this research is the direct estimation of the knee flexion angle from a single-view monocular RGB image.

In Section 1.1, we review SOTA pose estimation methods and group them into three categories: keypoint-based, orientation-based, and joint angle-based approaches. As detailed in Subsection 1.1.3, most existing methods rely on intermediate joint coordinate estimations; typically obtained from pretrained models such as OpenPose; to derive joint angles. This is done either through direct geometric calculations or by training a deep learning module that maps the coordinates into angles.

One notable exception is the work by Jamsrandorj *et al.* (2022) who built a LSTM-based model to directly estimate the flexion/extension angle. However, this approach relies heavily on temporal dependency and achieved a MAE of 5.15°, which is not considered clinically acceptable. To the best of our knowledge, no other existing approach that we know of opted for a direction angle estimation from monocular images.

To assess the feasibility of this direction and justify its relevance for our case study, (without temporal dependency) for our case study, we conduct an exploratory analysis to check whether the flexion angle an be reliably inferred from a lateral view image (Section 3.2). We then analyze the reliability of using an example of a pretrained pose estimation model to estimate the flexion angles of dataset acquired in a clinical dataset context (our case study) (Section 3.3).

3.1 Context and definitions

The experimental setup (refer to Chapter 4.1) involves two cameras: one positioned in the sagittal plane on the right side of the subject (parallel to their motion), and another placed frontally in the coronal plane. These two views capture, respectively, the right profile and the front view of the participant performing the required movement. With the help of the KneeKG system, reflexive markers, two RGB cameras and an infrared camera, we collect synchronized video sequences. Each frame is annotated with the 2D and 3D coordinates of key anatomical landmarks: the knee, ankle, and femoral head. In addition, each image is labeled with two anatomical coordinate frames: the Tibial Anatomical Frame (TAF) and the Femoral Anatomical Frame (FAF). These frames are expressed relative to the RGB camera coordinate system. Each anatomical frame is represented as a set of three orthonormal vectors with respect to the camera frame. The figure 3.1 shows an example of how both frames are oriented with the lower limbs. The red, green and blue vectors are respectively the orthonormal vectors X, Y and Z of each anatomical frame.

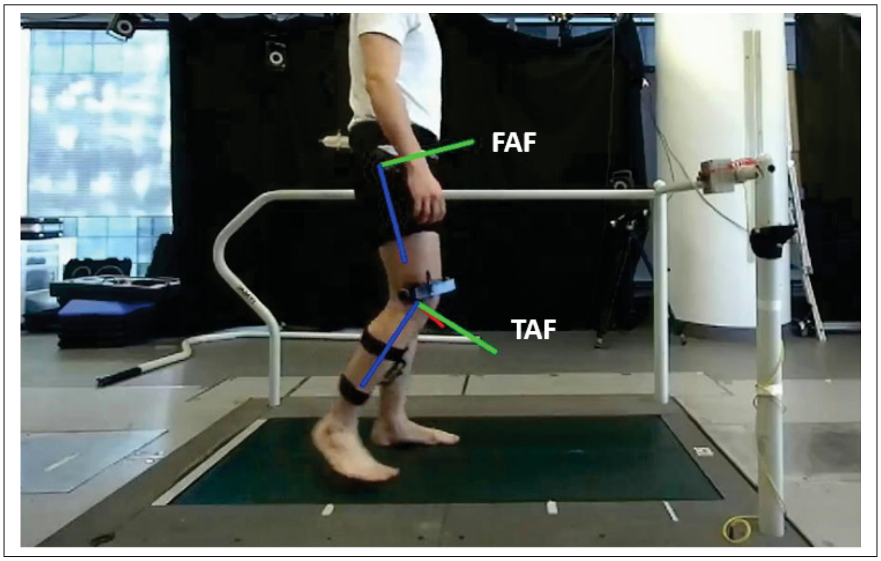


Figure 3.1 Anatomical frames visualization
FAF: Femur anatomical frame
TAF: Tibial anatomical frame

This is equivalent to a transformation from the camera's coordinate system (used as the reference or global frame) to the corresponding anatomical frame.

To describe the relationship between these coordinate systems, we use a 4×4 homogeneous transformation matrix, which includes both rotation and translation components (Deakin, 1999):

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{3.1}$$

where:

- $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix that defines the orientation of the new axes.
- $\mathbf{t} \in \mathbb{R}^{3\times1}$ is a translation vector that defines the new origin.

Applying this matrix to a coordinate frame changes:

- The **origin** — moved to a new position defined by **t**.
- The **axes directions** — reoriented by the rotation matrix **R**.

Since we are only interested in the relative orientation without the position of the axis origin, the orientation of each anatomical frame can be encoded as only the **R** matrix.

The transformation of coordinate axes $\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix}$ to another one $\boldsymbol{e'} = \begin{bmatrix} \boldsymbol{e}'_x & \boldsymbol{e}'_y & \boldsymbol{e}'_z \end{bmatrix}$ can be obtained using the rotation matrix $\mathbf{R}_{e\to e'}$ as follows:

$$\begin{bmatrix} \boldsymbol{e}'_x & \boldsymbol{e}'_y & \boldsymbol{e}'_z \end{bmatrix} = \mathbf{R}_{e\to e'} \begin{bmatrix} \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix} \tag{3.2}$$

where $\mathbf{R}_{e\to e'} \in \mathbb{R}^{3\times3}$ is the rotation matrix that aligns the original axes with the new axes. In our case $\begin{bmatrix} \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \end{bmatrix}$ is the canonical basis of the global (camera) frame and $\begin{bmatrix} \boldsymbol{e}'_x & \boldsymbol{e}'_y & \boldsymbol{e}'_z \end{bmatrix}$ is the basis of one anatomical frame.

The orientation of each anatomical frame is thus encoded as a 3×3 rotation matrix with respect to the global frame. The two rotation matrices relative to the femur and tibia are called $\mathbf{R}_{AF}$ and $\mathbf{R}_{AT}$. To express the orientation of the tibia with respect to the femur, we compute the relative rotation:

$$\mathbf{R}_{AF\to AT} = \mathbf{R}_{AF}^{-1} \times \mathbf{R}_{AT} \tag{3.3}$$

This relative rotation is used to extract clinically relevant angles: flexion/extension, abduction/adduction, and internal/external rotation, representing the motion in the sagittal, frontal, and transverse planes, respectively.

Emovi uses the inverse Euler Sequence XYZ to derive these anatomical angles from the rotation matrix. If we suppose that the rotation matrix is defined as:

$$\mathbf{R}_{AF\rightarrow AT} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{3.4}$$

The flexion/extension $\theta_{f/e}$, abduction/adduction $\theta_{a/a}$ and internal/external $\theta_{i/e}$ rotation angles are obtained by (Diebel, 2006):

$$\begin{pmatrix} \theta_{f/e} \\ \theta_{a/a} \\ \theta_{i/e} \end{pmatrix} = \text{euler}_{\text{XYZ}} = \begin{pmatrix} \theta_X \\ \theta_Y \\ \theta_Z \end{pmatrix} = \begin{pmatrix} \text{atan2}(r_{23}, r_{33}) \\ -\text{asin}(r_{13}) \\ \text{atan2}(r_{12}, r_{11}) \end{pmatrix} \tag{3.5}$$

where atan2 : $\mathbb{R} \times \mathbb{R} \longrightarrow [-\pi, \pi]$ is the four-quadrant inverse tangent:

$$\text{atan2}(y, x) = \begin{cases} \text{atan}(y/x) & \text{if } x > 0 \\ \text{atan}(y/x) - \pi & \text{if } x < 0 \text{ or } y < 0 \\ \text{atan}(y/x) + \pi & \text{if } x < 0 \text{ or } y > 0 \end{cases} \tag{3.6}$$

3.2 Angles and joints analysis

At this stage, the annotations include the 2D positions, the 3D coordinates and the three angles for each image. In previous works as stated in the literature review, many authors defined the flexion/extension angle as the angle geometrically formed by the tibia segment (the line between the knee and the ankle) and the femur segment (the line between the knee and the femoral head) from a sagittal plane view. For each image we compute the directed angles computed with simple geometric operations. The directed angle is defined by the angle from the 2D femoral axis vector to the 2D tibial axis vector as defined in the Figure 3.2 showing the lateral camera view.

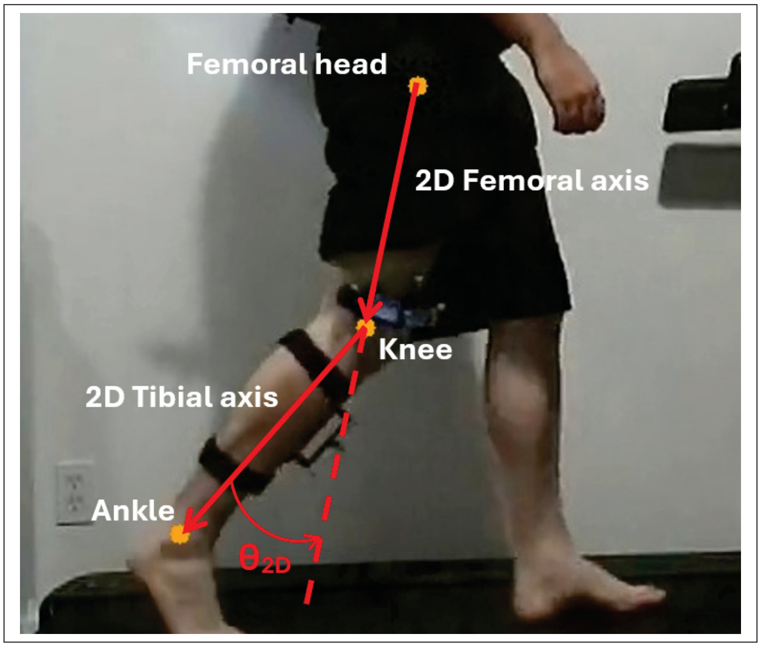


Figure 3.2 Computation of the 2D knee flexion angle from the lateral projected joint coordinates

We compute the correlation between the flexion/extension angles $\theta_{f/e}$ obtained from the Euler sequence and the geometrical angles θ_{2D}, evaluated on a per-trial, per-patient, and overall basis. All trial-level, participant-level and overall basis correlations show a value higher than 0.99. We show in the Figure 3.3 the correlation plots of the angles for two unique participants and the overall basis correlation plot.

The high correlations indicate that the geometric 2D angle closely approximates the anatomical flexion/extension angle, primarily because the movement is performed as parallel to the camera plane as possible.

We tried a similar analysis for the geometric 2D angle and the abduction/adduction angle to see if they correlate well. The figure below shows the 2D angle from a frontal view.

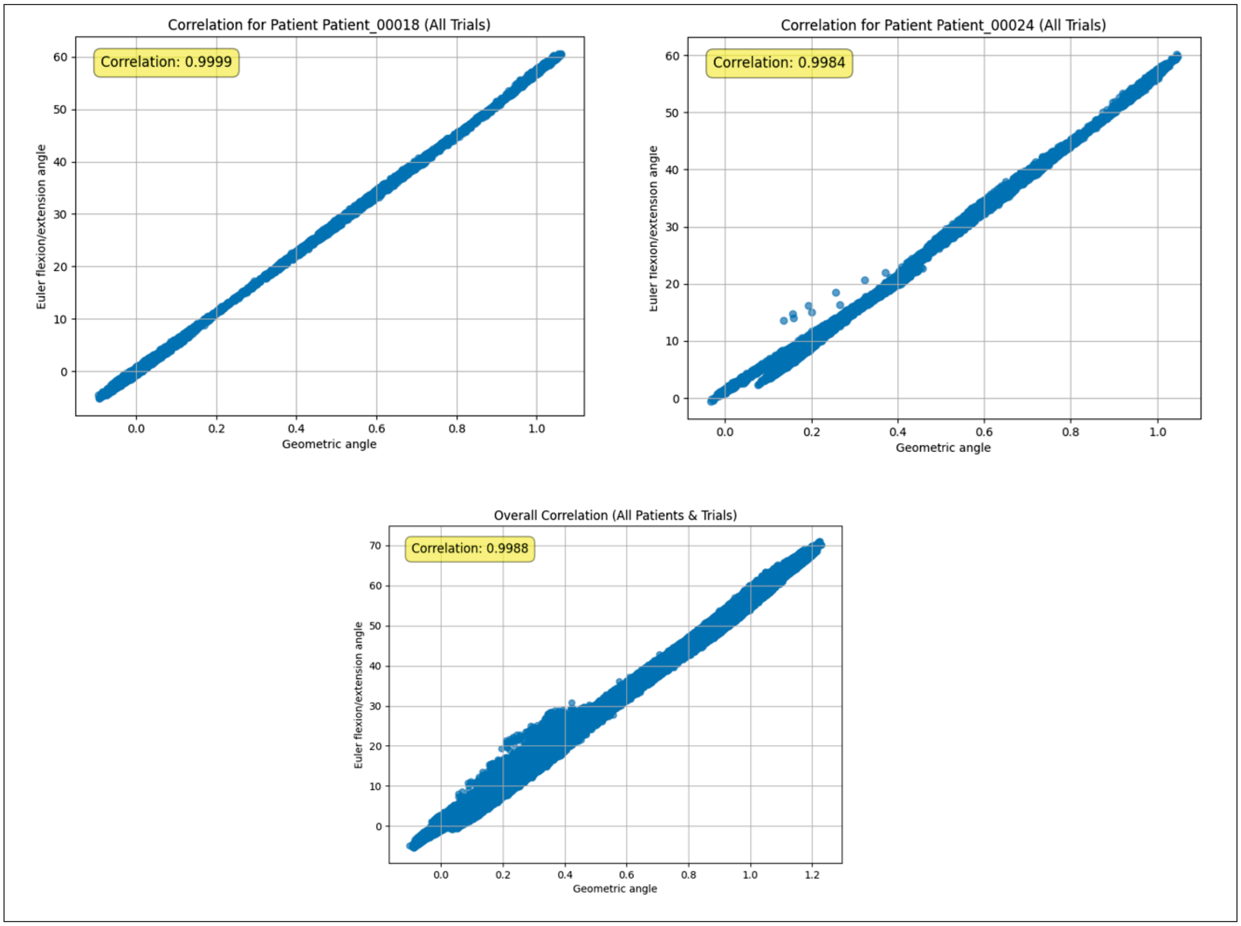


Figure 3.3 Correlation plots for two unique participants and for the whole dataset (lateral view)

The analysis show no correlation between the computed 2D angle and the abduction/adduction angle on trial, patient or global basis. In fact, the angles show a correlation coefficient of 0.14 on a global basis.

It is common in the literature to infer knee flexion angles from 2D lower-limb joint coordinates, as demonstrated in multiple prior studies (Kidzinski *et al.*, 2020; Le & Pham, 2024; Dinh *et al.*, 2025). However, the accuracy of angle estimation is heavily dependent on the precision of the underlying 2D joint coordinate predictions. While pose estimation models such as HRNet and OpenPose have demonstrated strong performance on large-scale benchmark datasets like COCO and MPII, their effectiveness may be reduced when applied to data collected in specific

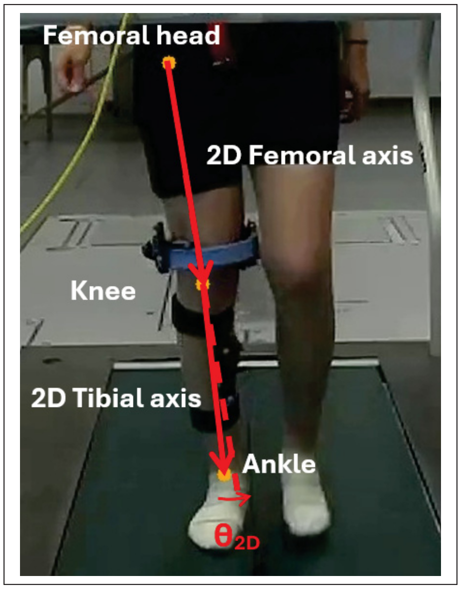


Figure 3.4 Computation of the 2D knee flexion angle from the frontal projected joint coordinates

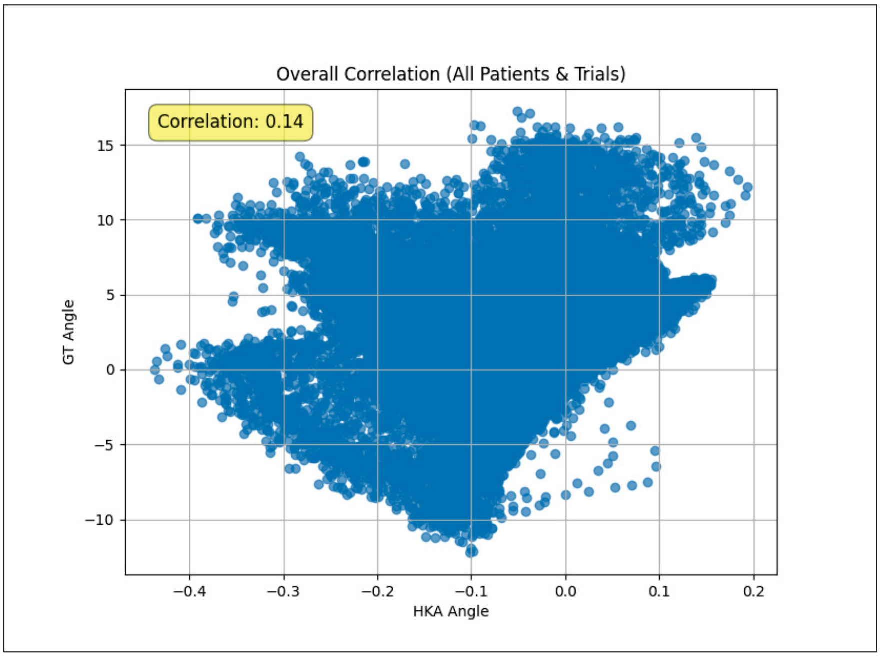


Figure 3.5 Correlation plots of the whole dataset (frontal view)

clinical settings, such as in the present study. In fact, public datasets such as COCO and MPII are typically annotated manually, without the aid of marker-based motion capture systems or clinically validated procedures. As a result, the joint annotations in these datasets may considerably deviate from those obtained using clinical-grade systems like the KneeKG. This discrepancy can lead to systematic differences in joint localization, and thus, builds up errors in estimating geometrically computed joint angles. Therefore, reliance on pretrained models developed on such datasets may introduce a significant source of error in clinical applications. In approaches that rely on using pretrained pose estimation models to compute joint coordinates as an intermediate step to infer joint angles, the overall reliability of the angle prediction pipeline is critically dependent on the accuracy of the joint coordinate predictions. Any inaccuracies and errors in the 2D pose estimation stage propagate and may result in substantial angular errors.

3.3 Practical analysis: HRNet

To evaluate this effect, we analyzed the performance of pretraining a pose estimation model HRNet (a reference and widely used model in HPE) on a single walking trial from one participant. Specifically, we assessed the model's 2D joints coordinates predictions of the three key joints involved in knee flexion. We used the *pose_hrnet_w48_384x288* model with 48 channels that was trained on COCO dataset with input resolution 384x288.
From the predicted joint positions, we computed the resulting geometric flexion angle and compared it to the ground truth obtained from the KneeKG. We report both the error statistics for the joint coordinates and the MAE for the inferred angles, thereby highlighting the impact of joint localization accuracy on the overall precision of angle estimation.

Table 3.1 Mean and standard deviation of joint coordinates (pixels px) and angle errors (degrees °) of the selected trial

Metric (px)	**Femoral Head Error** (px)	**Knee Error** (px)	**Ankle Error** (px)	**Angle Error** (°)
Mean (Std)	8.64 (4.59)	10.19 (14.36)	12.84 (22.19)	5.29 (9.71)

We observed high mean and standard deviation values in the joint location pixel errors, which led to correspondingly high angular errors and variability. These results confirm our claim that the pretrained models are not best suited for datasets obtained in clinical environments.

Conclusion

According to the literature review, the direct estimation of angles from monocular images without intermediate supervision such as keypoint coordinates remains an underexplored research direction with no clinically acceptable results reported so far. Based on that gap, this preliminary analysis served as motivation to look into the possibility of estimating the angle directly from the monocular source without any intermediate estimation. The high correlation observed between the projected 2D angle and the clinically computed flexion angle supports the assumption that the flexion angle is a reliably extractable feature from lateral-view images. The study also highlighted the limitation of pretrained pose estimation models in accurately estimating clinically computed joint coordinates, leading to compounded errors when the flexion angle is inferred geometrically.

In the next chapter, we explain the established data acquisition protocol that is sued to create this database.

CHAPTER 4

DATASET PREPARATION AND ACQUISITION PROTOCOL

Introduction

In this chapter, we explain how we set up the data acquisitions protocol and give a detailed overview about the obtained dataset.

4.1 Acquisition setup

In this section, we detail how the acquisitions were set up, including the required equipment and the available environments (Subsection 4.1.1). We then explain how the walking trials dataset was acquired from participants (Subsection 4.1.2).

4.1.1 Equipment and environment

The data acquisition is performed in one of the two separate environments:

- LIO laboratory: a biomechanical laboratory room in the seventh floor of the CRCHUM. Any participant is eligible to participate.
- EMOVI inc office: an isolated room with free space for trials. Only EMOVI workers and their clients are eligible to participate.

The necessary equipment and tools for the data acquisition are as follows:

Walking treadmill (Figure 4.2-a) : At the LIO laboratory, the walking is recessed into the laboratory floor. As for the Emovi office, a commercial-grade treadmill is used for the acquisitions. In case of the LIO trials, there is a software to launch and turn off the treadmill.

KneeKG : The KneeKG^{TM} tool includes several accessories as illustrated in Figure 4.1:

- A femoral harness (Figure 4.2-B) and a tibial plate **(C)** fixed quasi-statically on the right thigh and calf with attached passive motion sensors serving as a 3D tracker (Lustig *et al.*, 2012) with the reflective markers.
- A pelvis belt **(A)** with a pointer attached around the participant's pelvis.

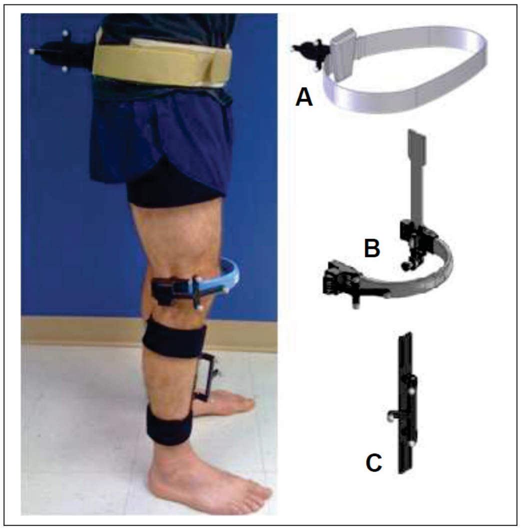


Figure 4.1 KneeKG^{TM} tool attached to a participant
Taken from Clément (2015)

Global frame tool (Figure 4.2-b) : A tool to detect the reflective markers positions. It serves as a reference frame for the motion capture.

T-shaped plate (Figure 4.2-c) : A T-shaped plate for patient orientation control during setup and calibration.

Video-acquisition system (Figure 4.2-d) Composed of two Stereolab Zed2i RGB-depth cameras mounted on height-adjustable tripods.

MoCap system (Figure 4.2-e) An infrared Polaris NDI camera to capture the reflective markers positions.

Computer (Figure 4.2-f) An installed computer with the Knee3D^{TM} (Emovi, Inc.) and the NDI Track softwares installed.

4.1.2 Data acquisition

There are multiple steps to complete an acquisition session. The detailed actions are in Appendix V.

Essentially, to perform an acquisition session:

1. Place the frontal and lateral stereolab Zed2i cameras, the Polaris NDI and the global frame tool in place.
2. The participant makes certain functional movements anatomical calibration (anatomical frames with respect to the technical frames)
3. The participant does two walking trials of approximately 45 seconds each.

The Figure 4.2 below shows a simplified representation of the key components of the experimental setup.

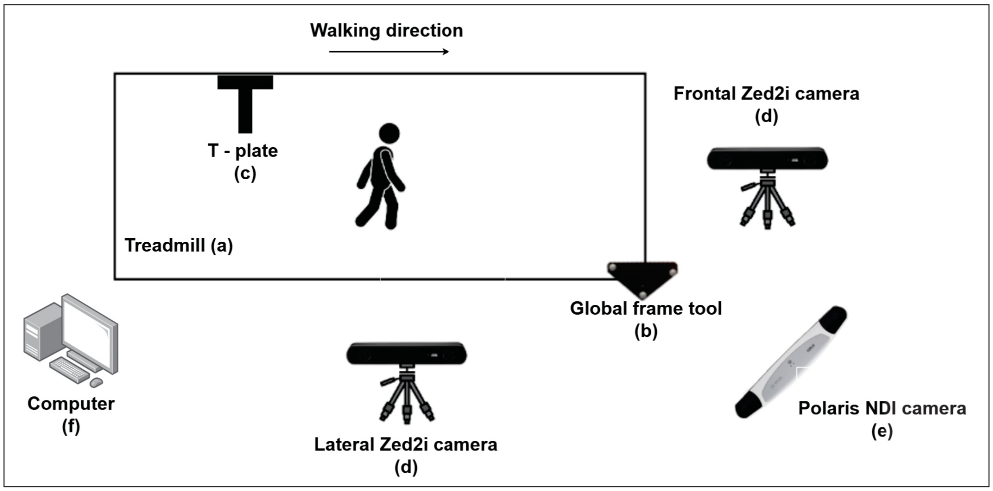


Figure 4.2 Simplified representation of the experimental setup

4.2 Raw dataset details

The collected data from the acquisition software and the cameras is saved across multiple key files:

- **Video sequences**: Each trial includes a RGB video file (`.mp4` format) recorded by the two Zed2i cameras. The videos are captured at 60 frames per second (fps) with a resolution of 1280×720 pixels.
- **Rigid body orientation sequences**: The orientations and time series of the technical frames attached to the femoral harness and tibial plate (rigid bodies) are stored in `.json` files. These files contain 4×4 transformation matrices (rotation and translation) representing the orientation of the femur and tibia technical frames (**TF** and **TT** respectively) relative to the global axis **GF** defined by the treadmill-attached tool, as tracked by the NDI camera system. They are respectively dented as $\mathbf{t}_{TF}^{GF}$ and $\mathbf{t}_{TT}^{GF}$. It also includes other irrelevant rigid bodies such as the pelvis belt.
- **Timestamps**: The frame timestamps corresponding to each Zed2i camera video are stored in a `.json` file. Each timestamp indicates the exact capture time for its associated frame. At 60 fps, the expected interval between two consecutive frames is approximately 16.66 ms, assuming frames are dropped. Another `.json` file contains the timestamps associated with each capture from the Polaris NDI camera.
- **Anatomical calibration data**: The anatomical landmarks positions and transformation matrices obtained during the participant's anatomical calibration procedure are stored in `.kkg` files. These files include:
  1. The lateral and medial condyles positions ($\mathbf{p}_{LC}^{TF}$ and $\mathbf{p}_{MC}^{TF}$), and femoral head position ($\mathbf{p}_{FH}^{TF}$) expressed in the technical femur frame (**TF**).
  2. The lateral and medial malleolus positions ($\mathbf{p}_{LM}^{TT}$ and $\mathbf{p}_{MM}^{TT}$) expressed in the technical tibia frame (**TT**).
  3. The transformation matrices representing the orientation from the technical femur frame (**TF**) to the anatomical femur frame (**AF**) denoted as $\mathbf{T}_{AF}^{TF}$, and from the technical tibia frame (**TT**) to the anatomical tibia frame (**AT**) $\mathbf{T}_{AT}^{TT}$. Both are stored as 4×4 matrices.
- **Camera intrinsic parameters**: The intrinsic parameters of each one of the Zed2i cameras, including focal lengths (f_x, f_y) and optical center coordinates (c_x, c_y), are stored in `.json` files. These parameters are used to project 3D camera space points into the 2D image plane.

- **Camera extrinsic parameters**: The extrinsic parameters represent the orientation and position of the camera frame (**CF**) relative to the global frame (**GF**). They are provided as 4×4 matrices and stored in `.json` format, denoted $\mathbf{M}_{CF}^{GF}$.

4.3 Data processing and preparation

Intermediate steps summarized in the Figure 4.3 are required to produce the final dataset used for model training and evaluation.

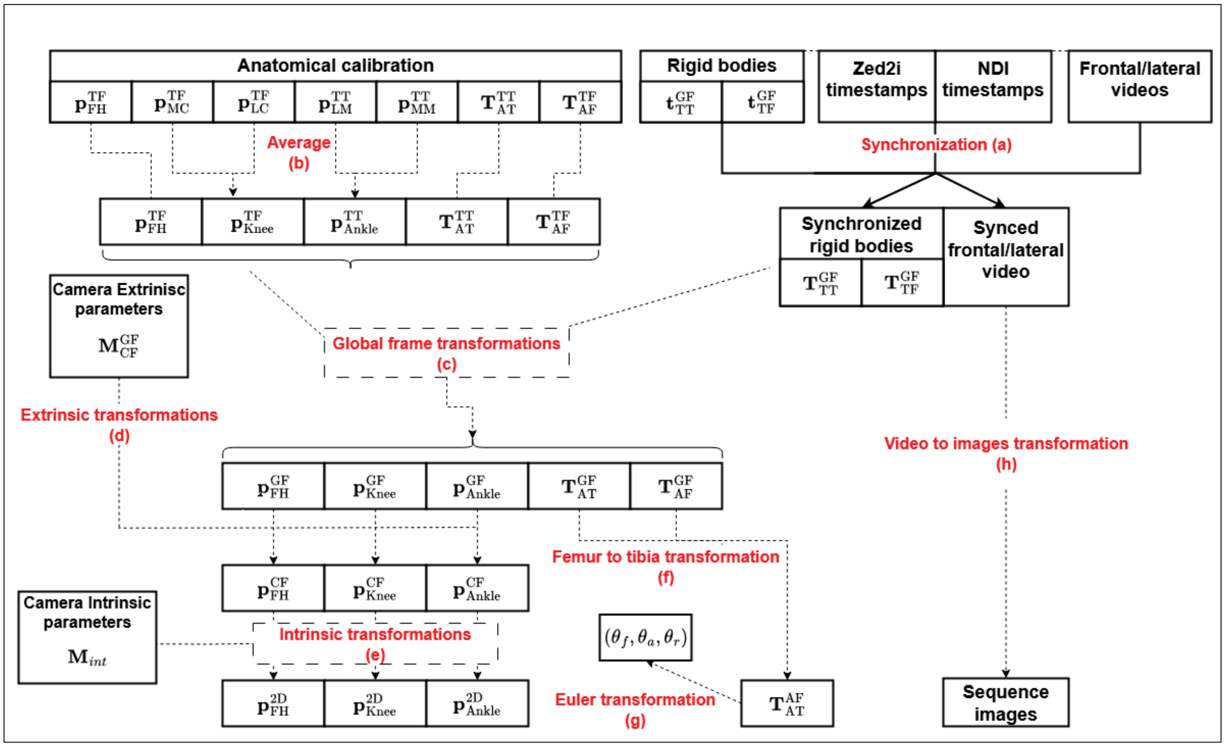

Anatomical calibration
$\mathbf{p}_{\mathrm{FH}}^{\mathrm{TF}}$
$\mathbf{p}_{\mathrm{MC}}^{\mathrm{TF}}$
$\mathbf{p}_{\mathrm{LC}}^{\mathrm{TF}}$
$\mathbf{p}_{\mathrm{LM}}^{\mathrm{TT}}$
$\mathbf{p}_{\mathrm{MM}}^{\mathrm{TT}}$
$\mathbf{T}_{\mathrm{AT}}^{\mathrm{TT}}$
$\mathbf{T}_{\mathrm{AF}}^{\mathrm{TF}}$
Rigid bodies
$\mathbf{t}_{\mathrm{TT}}^{\mathrm{GF}}$
$\mathbf{t}_{\mathrm{TF}}^{\mathrm{GF}}$
Zed2i timestamps
NDI timestamps
Frontal/lateral videos
Average (b)
Synchronization (a)
$\mathbf{p}_{\mathrm{FH}}^{\mathrm{TF}}$
$\mathbf{p}_{\mathrm{Knee}}^{\mathrm{TF}}$
$\mathbf{p}_{\mathrm{Ankle}}^{\mathrm{TT}}$
$\mathbf{T}_{\mathrm{AT}}^{\mathrm{TT}}$
$\mathbf{T}_{\mathrm{AF}}^{\mathrm{TF}}$
Synchronized rigid bodies
$\mathbf{T}_{\mathrm{TT}}^{\mathrm{GF}}$
$\mathbf{T}_{\mathrm{TF}}^{\mathrm{GF}}$
Synced frontal/lateral video
Camera Extrinisc parameters
$\mathbf{M}_{\mathrm{CF}}^{\mathrm{GF}}$
Global frame transformations (c)
Extrinsic transformations (d)
Video to images transformation (h)
$\mathbf{p}_{\mathrm{FH}}^{\mathrm{GF}}$
$\mathbf{p}_{\mathrm{Knee}}^{\mathrm{GF}}$
$\mathbf{p}_{\mathrm{Ankle}}^{\mathrm{GF}}$
$\mathbf{T}_{\mathrm{AT}}^{\mathrm{GF}}$
$\mathbf{T}_{\mathrm{AF}}^{\mathrm{GF}}$
Femur to tibia transformation (f)
$\mathbf{p}_{\mathrm{FH}}^{\mathrm{CF}}$
$\mathbf{p}_{\mathrm{Knee}}^{\mathrm{CF}}$
$\mathbf{p}_{\mathrm{Ankle}}^{\mathrm{CF}}$
Camera Intrinsic parameters
$\mathbf{M}_{int}$
Intrinsic transformations (e)
$(\theta_f, \theta_a, \theta_r)$
Euler transformation (g)
$\mathbf{T}_{\mathrm{AT}}^{\mathrm{AF}}$
Sequence images
$\mathbf{p}_{\mathrm{FH}}^{\mathrm{2D}}$
$\mathbf{p}_{\mathrm{Knee}}^{\mathrm{2D}}$
$\mathbf{p}_{\mathrm{Ankle}}^{\mathrm{2D}}$

Figure 4.3 Transformation of raw data to useful features

First, we extract the relevant rigid body transformation matrices: $\mathbf{T}_{TF}^{GF}$ and $\mathbf{T}_{TT}^{GF}$ correspond to the femoral harness and tibial plate, respectively. The two Zed2i cameras and the NDI camera timestamps are synchronized (Figure 4.3-a). To align both data sources, the timestamps from the two Zed2i cameras and the Polaris NDI camera are synchronized (Figure 4.3-a). This synchronization step addresses potential issues such as dropped frames or slight discrepancies in camera start times. We obtain synchronized video sequences, which are then converted into image sequences through frame-by-frame extraction (Figure 4.3-h). Similarly, we obtain the synchronized rigid body transformations $\mathbf{T}_{TF}^{GF}$ and $\mathbf{T}_{TT}^{GF}$.

Next, the 3D coordinates of the knee and ankle are obtained by averaging respectively the coordinates of the lateral and medial condyles, and the lateral and medial malleolus (Figure 4.3-b).

$$\begin{aligned} \mathbf{p}_{Knee}^{TF} &= \frac{\mathbf{p}_{MC}^{TF} + \mathbf{p}_{LC}^{TF}}{2} \\ \mathbf{p}_{Ankle}^{TT} &= \frac{\mathbf{p}_{LM}^{TT} + \mathbf{p}_{MM}^{TT}}{2} \end{aligned} \tag{4.1}$$

Then, using the synchronized technical frames transformations with respect to the global axis **GF** ($\mathbf{T}_{TF}^{GF}$ and $\mathbf{T}_{TT}^{GF}$), the knee, ankle and femoral head positions with respect to their technical frames ($\mathbf{p}_{knee}^{TF}$, $\mathbf{p}_{Ankle}^{TT}$, $\mathbf{p}_{FH}^{TF}$) and the technical to anatomical frames tibial and femur transformations ($\mathbf{T}_{AT}^{TT}$ and $\mathbf{T}_{AF}^{TF}$), we obtain the same coordinates with respect to the global axis (GA) and the same transformations relative to the global axis (Figure 4.3-c):

$$\begin{aligned} \mathbf{p}_{FH}^{GF} &= \mathbf{T}_{TF}^{GF} \cdot \mathbf{p}_{FH}^{TF} \\ \mathbf{p}_{Knee}^{GF} &= \mathbf{T}_{TF}^{GF} \cdot \mathbf{p}_{Knee}^{TF} \\ \mathbf{p}_{Ankle}^{GF} &= \mathbf{T}_{TT}^{GF} \cdot \mathbf{p}_{Ankle}^{TT} \\ \mathbf{p}_{AT}^{GF} &= \mathbf{T}_{TT}^{GF} \cdot \mathbf{p}_{AT}^{TT} \\ \mathbf{p}_{AF}^{GF} &= \mathbf{T}_{TF}^{GF} \cdot \mathbf{p}_{AF}^{TF} \end{aligned} \tag{4.2}$$

Using the camera extrinsic parameters $\mathbf{M}_{CF}^{GF}$, we transform the 3D points from the global coordinate system to the camera coordinate frame (Figure 4.3-d):

$$\begin{aligned} \mathbf{p}_{FH}^{CF} &= \left(\mathbf{M}_{CF}^{GF}\right)^{-1} \cdot \mathbf{p}_{FH}^{GF} \\ \mathbf{p}_{Knee}^{CF} &= \left(\mathbf{M}_{CF}^{GF}\right)^{-1} \cdot \mathbf{p}_{Knee}^{GF} \\ \mathbf{p}_{Ankle}^{CF} &= \left(\mathbf{M}_{CF}^{GF}\right)^{-1} \cdot \mathbf{p}_{Ankle}^{GF} \end{aligned} \tag{4.3}$$

Using the camera intrinsic matrix $\mathbf{M}_{\text{int}}$, we project each 3D point $\mathbf{p}^{3D} = [X, Y, Z]^T$ expressed in the camera coordinate system into 2D pixel coordinates (Figure 4.3-e). We use the camera intrinsic matrix $\mathbf{M}_{int}$:

$$\mathbf{M}_{int} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

The projection into the pixel coordinates is given by:

$$\mathbf{p}^{2D} = \mathbf{M}_{int} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x \cdot (X/Z) + c_x \\ f_y \cdot (Y/Z) + c_y \\ 1 \end{bmatrix} \tag{4.5}$$

The corresponding 2D image coordinates (x, y) are:

$$x = f_x \cdot \frac{X}{Z} + c_x, \quad y = f_y \cdot \frac{Y}{Z} + c_y \tag{4.6}$$

Using this procedure, we obtain the 2D image coordinates of the femoral head, knee and ankle joints ($\mathbf{p}^{2D}_{FH}$, $\mathbf{p}^{2D}_{Knee}$, $\mathbf{p}^{2D}_{Ankle}$).

Finally we compute the femur to tibia transformation $\mathbf{T}^{AF}_{AT}$ using both tibial and femur transformation matrices (Figure 4.3-f).

$$\mathbf{T}^{AF}_{AT} = \left(\mathbf{T}^{GF}_{AF}\right)^{-1} \cdot \mathbf{T}^{GF}_{AT} \tag{4.7}$$

Subsequently, we extract the three Euler angles: flexion, abduction and rotation (θ_f, θ_a and θ_r) through the XYZ sequence decomposition (Figure 4.3-g).

The final relevant components of the dataset which will be used in both the methodological development and the evaluation phases, are as follows:

- The image frames: The individual RGB images extracted at 60 fps from the synchronized video sequences.
- The joints (Femoral head, knee and ankle) 2D positions: p^{2D}_{FH}, p^{2D}_{Knee} and p^{2D}_{Ankle}
- The knee joint flexion angle, θ_f. For simplicity reasons, we explicitly denote it as θ.

- The femur to tibia transformation $\mathbf{T}_{AT}^{AF}$. The transformation is a 4×4 homogeneous transformation matrix. Since only the relative orientation (rotation) is of interest, we retain only the 3×3 rotation submatrix, and we call it $\mathbf{R}$.

After we finished all the acquisitions and data processing, we ended up with a ready-to-use database. The table 4.1 summarizes the number of available trials and participants for each view.

Table 4.1 Summary of walking trial and participant counts for different camera views

View	**Repartition**	**Count**
Lateral	Trials	62
	Participants	30
Frontal	Trials	24
	Participants	13

Some walking trials have occasional synchronization issues or temporary loss of reflective marker tracking. This causes certain frames of these trials to be dropped. Among the 62 considered trials, the length of the shortest one is 1485 frames and the longest 2699 frames, with an average length of 2520 frames.

According to the preliminary study in Chapter 3, only the flexion angle is likely to be inferred from the lateral view images; but the abduction angle is very unlikely to be explicitly predicted from the frontal view images. In the next chapters, we focus on the lateral view dataset only.

4.4 Mean gait cycle analysis

In this section, we analyze the phases of a mean NGC and its phases for a walking trial. We take a trial of one participant and apply the mean gait cycle normalization procedure:

1. Extract the heel strike localization (indexes in the time series) of all the trials by looking for the local maxima of the $p_{Ankle}^{2D} - p_{FH}^{2D}$ values.
2. Between each two heel strikes, there is a gait cycle of n_c points.

3. Apply a spline interpolation function to interpolate the angle values over the extracted cycle to a normalized 0-100 space:. This results in a normalized cycle represented by 101 uniformly spaced values (corresponding to integer percentages). The obtained normalized gait cycle is called percentage gait cycle.
4. All the obtained percentage gait cycles are averaged point by point to obtain a mean percentage gait cycle.

The obtained NGC is presented in Figure 4.4. The figure highlights the key phases of the cycle.

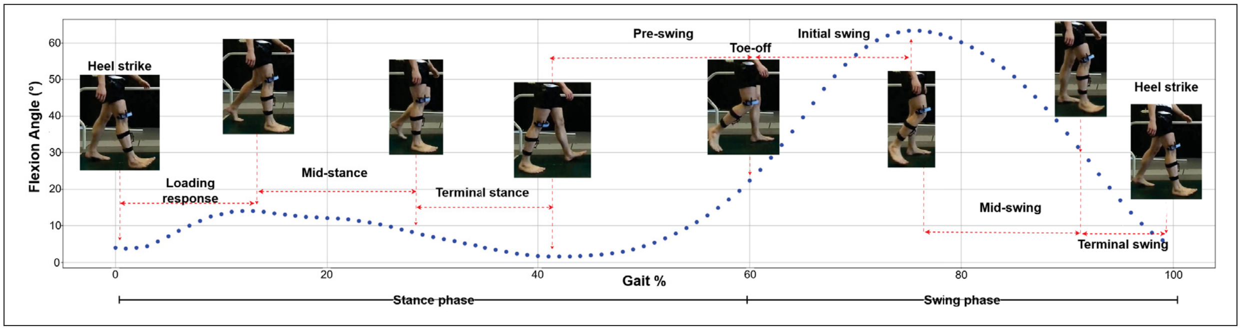


Figure 4.4 Normalized Mean gait cycle plot of one trial

The flexion angles are at their lowest by the start of the stance phase and the end of the swing phase. Their peak is reached by the end of the stance phase and the start of the swing phase.

Conclusion

In this chapter, we described the process of constructing the dataset, from defining the acquisition protocol to extracting the final relevant information. We also provided details on the dataset size and the temporal structure of the trials time series by presenting the mean gait cycle phases.

In the next chapter, we present the methodology designed to address the research objectives.

CHAPTER 5
METHODOLOGY

Introduction

This chapter presents the methodological pipeline adopted to investigate the research questions posed in this study. It details the model architecture, the training and validation strategies and the data processing steps. The methodological approach was motivated by the conclusions from the analysis and study of Chapter 3 and the literature review of Chapter 1, which aim to directly estimate knee joint flexion angles from monocular RGB images using deep learning. The chapter is organized as follows: Section 5.1 describes the pipeline structure and details its components including the data preparation and pre-processing steps, the network modules, the loss functions and the constraints. Section 5.2 detail the steps to fine-tune and train the model. Section 5.4 show enumerate steps of the final model evaluation. In Section 5.3 we explain how we train and test the other SOTA approaches and how to compare their performance with our approach.

5.1 Pipeline definition

The Figure 5.1 illustrates the overall pipeline starting from raw data acquisition, followed by data pre-processing, deep learning model training and inference, and ending with the final evaluation.

The first step is data acquisition (**a**). This is the first step of the pipeline and it includes doing the experiments and processing the obtained data. The data preparation process is described in Chapter 4. The processed dataset contains images and their annotations: the three lower joint coordinates and the flexion angles. The acquired data is then passed through a dataset split process (**b**) detailed in 5.1.1. An image inpainting module (**c**) (Subsection 5.1.2) removes the KneeKG from each image and creates an identical dataset with the same number of images and the same annotations. We refer to the dataset before image inpainting as the standard dataset, and the one after inpainting as the inpainted dataset. The BB coordinates are obtained through the BB module (**d**) and result in a cropped image with only the region of interest. Both

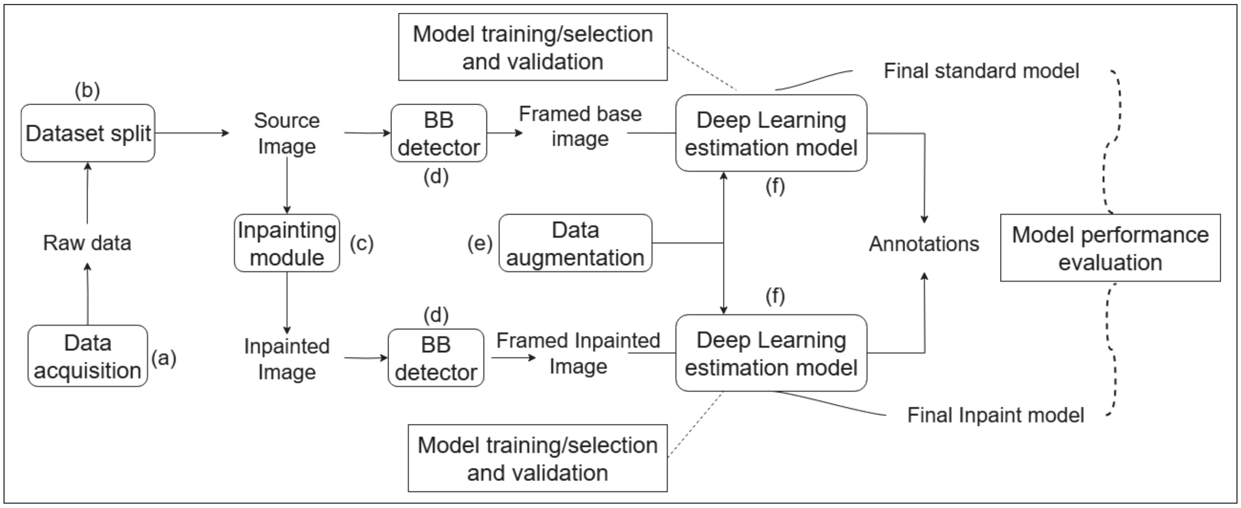


Figure 5.1 Pipeline overview

datasets (cropped standard and cropped inpainted datasets) are used to train and validate two separate models: one model for each dataset using the deep estimation model (**f**) explained in Subsection 5.1.5. The data used for training undergoes data augmentation module (**e**) explained in the Subsection 5.1.4.The performance of the final model for each dataset is then evaluated.

5.1.1 Dataset split

On the highest level, the dataset is composed of N_{total} participants (patients) with each one performing walking trials. The details of the dataset and how it was acquired are in the chapter 4 In this project, we use CV to select, train, validate and evaluate our model. We use the test-holdout Group Repeated CV as explained in 1.3.1 due to its capacity to reduce the data-related bias in hyperparameter selection. The data is randomly split based on the patients (no overlap: each one is only used for one set). The test set is composed of n_{test} participants. The rest of the dataset is used for CV and split to K folds with n different random split combinations: We split the remaining patients to K folds with no overlap with one fold selected each time for validation and the other ones for training. We repeat this process n times resulting in $K \times n$ total train-validation split combinations of the dataset with the final test set always set apart.

5.1.2 Image inpainting

The image inpainting is performed using the LaMA inpainting model and with the help of an internally developed tool that uses an interactive GUI. The tool serves to observe the result of the KneeKG detection before applying the inpainting process by LaMa.

For each trial, the first image is selected and three initial BB estimates are generated to locate the KneeKG system and its accessories. The estimation are approximations of their locations based on the joints coordinates. For example, the center of the green BB is positioned at the midpoint of the tibia segment (the line between the knee and ankle joints).

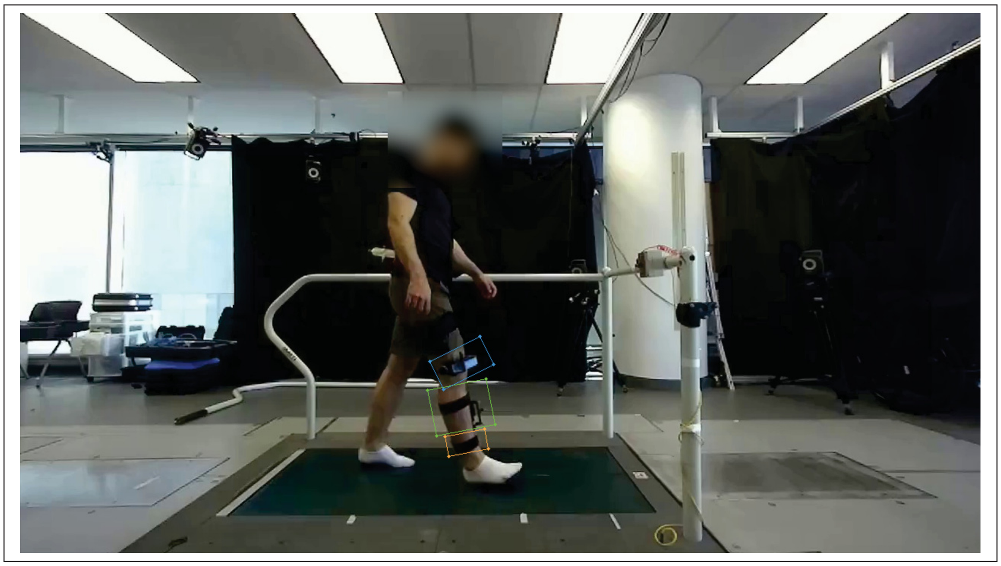


Figure 5.2 Initial KneeKG BB estimation

The initial estimations may not be sufficiently accurate and require more adjustment of the BB padding—that is, the extent to which the BBs are extended—to ensure that they dully cover the the KneeKG accessories while avoiding the inclusion of irrelevant image regions. The GUI allows to visualize the BBs across the entire video, making it possible to iteratively fine-tune the padding settings. To ensure consistent inpainting accuracy throughout the sequence, the adjustments can be validated over the initial gait cycle (roughly the first 50 to 70 frames).

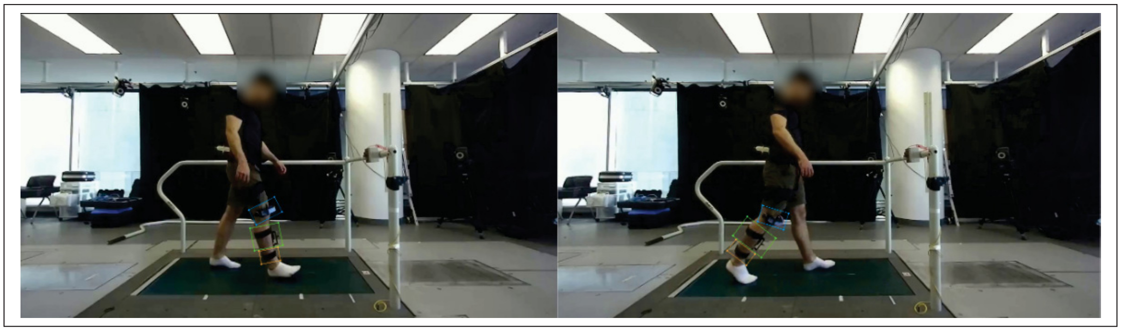


Figure 5.3 The KneeKG parts BB detector parameters adjusted to different frames of the trial

Once the three BBs have been appropriately refined, the KneeKG regions can be inpainted properly in each frame. The LaMa model processes each frame independently, using the masked regions to produce as close to real as possible inpainted images.

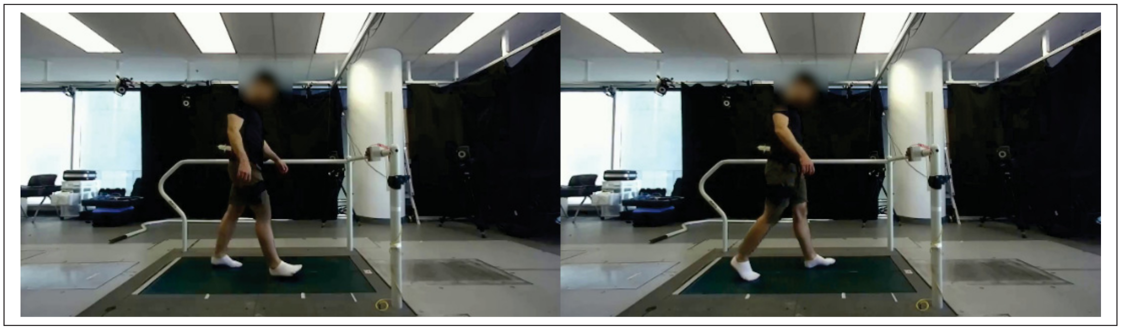


Figure 5.4 Frames inpainted by LaMa

This procedure is repeated for all the trials of the dataset. We obtain a new inpainted dataset with the KneeKG removed from the images but with the same annotations (joint coordinates and angles) and image size.

5.1.3 Lower body bounding box detector

HPE models typically rely on full-body BB detectors because they aim to estimate the whole skeleton joints. In this case, the region of interest only includes the lower body, specifically the segments involved in motion analysis. The purpose of BB detection in this context is to

eliminate irrelevant image regions, particularly background areas or body parts not contributing to the estimation task in order to reduce noise and enhance the focus on relevant features.
Since there are no specific lower body detectors that we know of, we opted to construct a compound detector that locates the full-body BB using the real-time object detector RTMDet (Lyu *et al.*, 2022) (trained on coco dataset) (**a**) then use a pretrained OpenPose estimator that estimates both the right and left femoral head 2D joints coordinates (**b**). These two joints serve as reference points to accurately crop the lower-body region from the full-body BB (**c**).
An example is illustrated in the Figure 5.5:

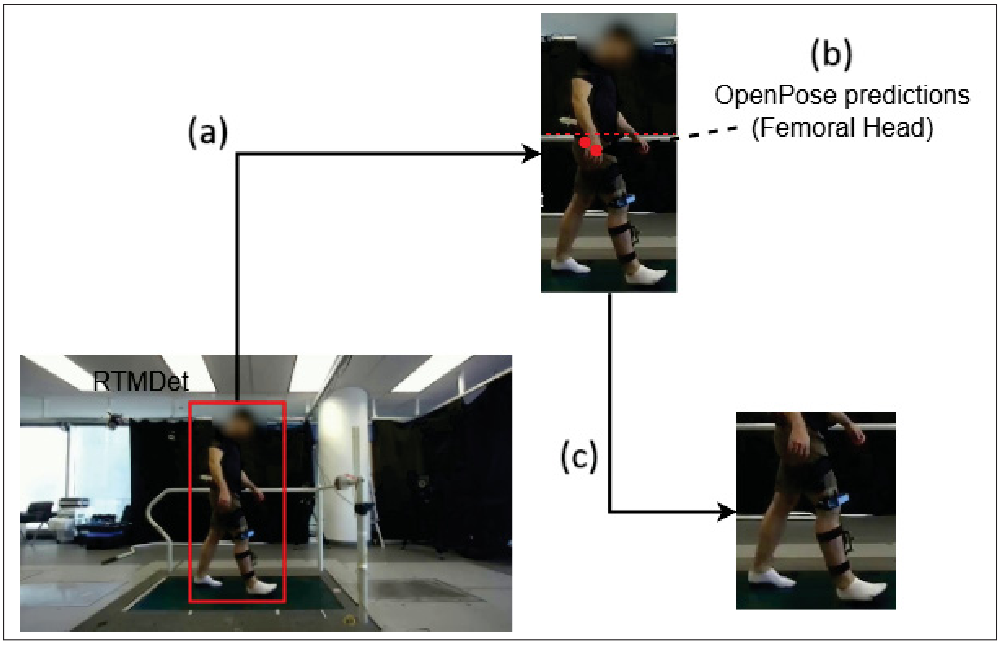


Figure 5.5 Lower body BB detection pipeline

5.1.4 Data augmentation

For data augmentation, we visually experimented with different possible image augmentations inspired from the literature review (Subsection 1.4.1) and chose their magnitude in order to avoid generating unrealistic samples.
The dataset is acquired at roughly the two same environments with a fixated horizontal camera at the same spot parallel to the person's sagittal plane. Extreme augmentation is not very useful

since there is no variety in the camera angles or perspectives in the train set compared to the test set. We limit the image augmentations to a rotation between $-5°$ and $5°$ and a zoom between 80% and 120%. The transformations are directly applied to the image and the joints coordinates are adjusted accordingly, as they are required to generate the feature maps. The angles, however, remain unaffected by these transformations.

5.1.5 Deep learning model architecture

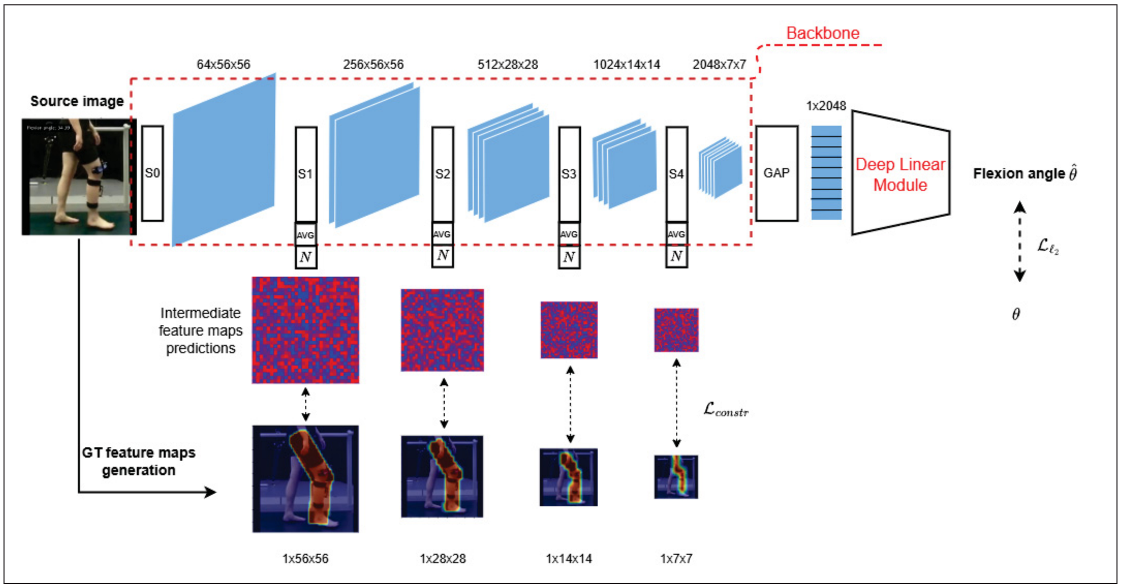


Figure 5.6 Deep learning model architecture

The proposed deep learning architecture consists of a convolutional backbone based on ResNet-50 or RepVGG-B1G2 composed on an initial stage $S0$ and four intermediate stages from $S1$ to $S4$, followed by a deep linear regression head that outputs a single scalar value: the predicted knee flexion angle. The model incorporates intermediate supervision with feature maps extracted after each stage of the backbone, as illustrated in Figure 5.6.

The following paragraphs describe each part of the model in detail.

5.1.5.1 Backbone

The backbone is based on the stem stage S0 and the four Convolutional stages S1, S2, S3 and S4 of ResNet50 or RepVGG-B1G2. These stages are separated to highlight the extraction and supervision of intermediate feature maps, promoting more targeted and interpretable feature learning. For the ResNet architecture, the four stages composed of multiple bottleneck blocs with different output channels dimensions. Similarly, RepVGG contains multiple VGG-style blocs. The output channels dimensions (respectively $64 \times 56 \times 56$, $256 \times 56 \times 56$, $512 \times 28 \times 28$, $1024 \times 14 \times 14$ and $2048 \times 7 \times 7$) are mentioned in the Figure 5.6. Each stage produces a larger number of smaller feature maps. The deeper layers of convolutional networks extract deeper, more abstract and semantic interpretable features than the prior layers (Zeiler & Fergus, 2014).

At the end of each stage, a ReLU activation is applied to introduce non-linearity and avoid the risk of vanishing gradients (Goodfellow *et al.*, 2016; Glorot, Bordes & Bengio, 2011). However, to retain more of the raw features information, we extract the feature maps before the ReLU activation. Two processing steps are then applied:.

- Channel-wise Averaging (AVG): For each spatial location (i, j) across all channels k, we compute the pixel-wise mean. This transforms the $K \times H \times W$ feature maps tensor to a single $H \times W$ 2D feature map by averaging across the channel dimension.
- Min-Max Normalization ($\mathcal{N}$): The averaged feature map is scaled to the range $[0, 1]$, producing a normalized feature map $\hat{F}$ with each pixel value reflecting a probabilistic relevance score. The more feature map pixel value is closer to 1, the more the associated pixel in the image is likely to be relevant to the estimation.

$$\mathcal{N}(F)(i,j) = \frac{F(i,j) - \min\limits_{(i,j)\in\{1,\dots,H\}\times\{1,\dots,W\}} F(i,j)}{\max\limits_{(i,j)\in\{1,\dots,H\}\times\{1,\dots,W\}} F(i,j) - \min\limits_{(i,j)\in\{1,\dots,H\}\times\{1,\dots,W\}} F(i,j)} \tag{5.1}$$

Mapping the values between $[0, 1]$ serves as a probabilistic representation of those regions. Since we want the model to focus on the right lower limb of the person, guiding the model to focus on the pixels surrounding the leg helps the model focus more on the relevant regions. We consider this method as a type of regularization or an intermediate supervision.

As an example, we detail in Figure 5.7 the added feature extraction process to stage 2 of ResNet, which consists of four bottleneck blocks. Feature maps are extracted and processed at this level to demonstrate the proposed mechanism. The continuous lines represent the typical ResNet stage forward propagation while the discontinued lines are the added intermediate feature maps constraints propagation.

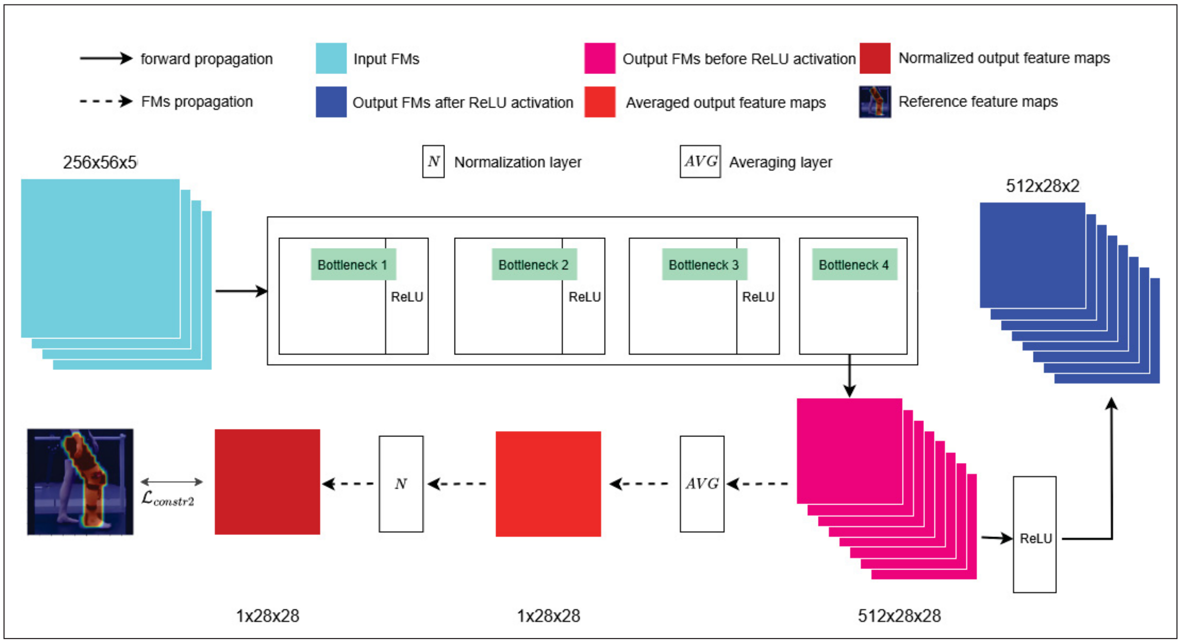


Figure 5.7 The modified ResNet stage 2 detailed architecture

Between every two stages, we add a dropout layer of rate $d = 0.3$ to reduce the overfit.

5.1.5.2 Deep linear regression module

The features obtained after the ReLU activation of the last stage are aggregated through a global average pooling (GAP) layer to obtain a one-dimensional vector of size 2048.

In order to map the feature vector to the flexion angle, we construct a deep linear regression module composed of two hidden linear layers and one final linear layer. We add a Leaky ReLU activation function and a batch normalization layer to the first two hidden layers to respectively avoid vanishing gradients (Maas, 2013) and accelerate training (Ioffe & Szegedy, 2015).

5.1.6 Loss function, constraints and regularization definition

Main objective and constraints

The model is primarily optimized using the main objective function, the mean squared error (MSE), denoted as $\mathcal{L}_{\ell_2}$:

$$\mathcal{L}_{\ell_2} = \|\hat{\theta} - \theta\|_2^2 \tag{5.2}$$

The angle is expressed in degrees.

In addition to this primary loss, we add 4 auxiliary constraints applied to the intermediate feature maps to guide the model in its training. Specifically, for each convolutional stage Si where i in $\{1, 2, 3, 4\}$, an additional ℓ_2 loss term is introduced to favor the predicted feature maps $F\hat{M}_i$ to align with manually constructed reference feature maps FM_i constructed from the source images (feature maps generation explained in 5.6). These auxiliary terms serve as network regularizers , encouraging the network to extract semantically relevant features at the four different stages.

The intermediate feature maps are also optionally optimized to match the manually generated feature maps from the source image. A ℓ_2 loss term for each stage from 1 to 4 is added to the total loss function. These regularization losses aim to guide the feature selection of the model at different stages of the backbone.

$$\mathcal{L}_{const_i} = \|F\hat{M}_i - FM_i\|_2^2 \tag{5.3}$$

where both feature maps of the same dimension $(1, H_i, W_i)$.
For ResNet and RepVGG the dimensions are detailed in Figure 5.6.

The ground-truth feature maps (pseudo-labels) generation

We generate the pseudo-labels from the ground-truth data which consists of the image and the 2D positions of the three lower limbs joints (knee, ankle and femoral head). The idea is to select the area around the pixels of the image that cover the right leg of the person and interpolate to the feature map resolution.

The process of creating the feature maps is outlined below and illustrated in the next Figure 5.8:

- A line is drawn between the femoral head and the knee joint and another one between the knee and ankle (**a**).
- A rectangle is constructed around this segment, with a length equal to the segment plus an additional 10 pixels on both the top and bottom sides. The rectangle is oriented such that its longer side is parallel to the segment (**a**). A gaussian blur is added to the rectangle t smoothen the edges of the rectangles in the features maps. The illustrative maps are colored to mark the covered active region. The red color signifies a pixel value equal to 1 and the rest colored as blue mean a null value.
- The two rectangles corresponding to the femoral head–knee and knee–ankle segments are combined to form the right lower leg feature map at a resolution of $(224, 224)$ (**b**).
- The feature maps are down-sampled to match the corresponding resolutions of the different backbone stages (**c**). The more advanced the stage, the lower the resolution and the less detailed and sharp the rectangles become and the more they lose their fine structure.

The compound cost function

After defining the loss terms, we can define the final cost function that is based on a combination of all the terms.

$$\mathcal{L} = \mathcal{L}_{\ell_2} + \sum_{i=1}^{4} \lambda_i \cdot \mathcal{L}_{\text{constr}_i} \tag{5.4}$$

This term will be used to train the model on predicting the flexion angle.

5.2 Training/Validation steps

To rigorously and reliably evaluate the performance of the pipeline, multiple steps are undertaken. After the data was acquired, processed and inpainted, the model is ready for training and validation. It is important to mention that we train two distinct versions of the model: one using the standard dataset containing the KneeKG system within the images and the other one on the inpainted dataset. The following subsections describe the process for each dataset separately.

Hyperparameters

Hyperparameters related to the model architecture or training settings are either fixed or tuned by CV.

The fixed hyperparameters are:

- The learning rate is fixed to 1e−4. We tried multiple learning rates with the 12 CV combinations and noticed no significant difference in the validation performance with different convergence rates. We chose 1e−4 for the learning rate as it offers a good trade-off between convergence speed and stability.
- Optimizer: ADAM optimizer.
- Batch size is fixed to 80.
- Input normalization: The image tensors are typically normalized by the pretrained ImageNet weights. This is a common useful practice in deep learning to normalize the images pixels to a standard normal distribution for a faster training convergence. However, since our unnormalized images may have a different distribution, we verify if those weights are still valid. A random training set of one of the split configurations is selected and all its images are collected. Using the lower BB cropped images, the mean and standard deviation weights of the per-channel images pixels are computed. Specifically, for each RGB channel $c \in \{R, G, B\}$, we compute the mean μ_c and standard deviation σ_c:

$$\mu_c = \frac{1}{N}\frac{1}{H \times W}\sum_{i=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W} I_{i,c}(h, w) \tag{5.5}$$

$$\sigma_c = \sqrt{\frac{1}{N}\frac{1}{H \times W}\sum_{i=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W} \left(I_{i,c}(h, w) - \mu_c\right)^2} \tag{5.6}$$

where:

N is the total number of images in the dataset,

H and W are the height and width of each image,

$I_{i,c}(h, w)$ is the pixel value at location (h, w) in channel c of image i,

The pixel values are assumed to be in the $[0, 1]$ range (original pixel values divided by 255).

We compute (μ_c^E, σ_c^E) and (μ_c^L, σ_c^L) relative to Emovi images and LIO images separately and we compare them to the pretrained ImageNet weights (μ_c^I, σ_c^I).

Table 5.1 Image channel-wise mean (μ) and standard deviation (σ) for EMOVI, LIO, and ImageNet datasets

	EMOVI (μ^E, σ^E)	**LIO** (μ^L, σ^L)	**ImageNet** (μ^I, σ^I)
μ	[0.43, 0.42, 0.40]	[0.28, 0.28, 0.25]	[0.485, 0.456, 0.406]
σ	[0.24, 0.25, 0.25]	[0.22, 0.20, 0.19]	[0.229, 0.224, 0.225]

Both LIO and Emovi weights are different from the ImageNet values. It is best to compute the image normalization statistics for all the samples of each environment (Emovi or LIO) before the start of each training.

- The data augmentations are only applied to train sets and include random rotations and zooms according to the parameter values specified in Section 5.1.4.
- The maximal number of epochs is set to 50 epochs as the model training curve stabilizes early in training and no significant validation improvement was observed beyond this point.

The hyperparameter that will be tuned by CV (Subsection 5.2) are:

- The backbone: ResNet50 or RepVGG-B1G2 backbones. Both models produce intermediate features with the same output's dimensions and share the input resolution of $(3\times224\times224)$ of ResNet50 and its equivalent RepVGG-B1G2 are the same. Consequentially, the architecture of the deep linear module is kept identical for both backbones. We use ImageNet pretrained versions of each backbone to benefit from transfer learning.
- The constraints loss weighting coefficients are denoted λ_k where $k \in \{1, 2, 3, 4\}$ corresponds to the intermediate stage index. For example, if $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.$, the training is done without any constraint included and $\mathcal{L} = \mathcal{L}_{\text{l2}}$.

Through gradient analysis, we observed that the magnitude of the constraint loss gradient is approximately 1000 times smaller than that of the main loss. To manage a reasonable amount of training runs, we limit each λ_k value to be in $\{0, 1000\}$, reducing the total number of constraint hyperparameter combinations to $2^4 = 16$.

Cross-validation and training

The dataset is divided into a separate test and a big train set consisting of respectively n_{test} and n_{train} participants as described in Subsection 5.1.1. Each shuffle split is denoted SS_i, where $i \in \{1, .., n\}$ and the corresponding folds as $F_{i,j}$, where $j \in \{1, .., K\}$. For each split SS_i, the model is trained using a unique hyperparameter combination across four different training/validation splits. Each configuration involves training on three folds and validating on the remaining fold, ensuring that each participant's data is used for training three times and validated once per shuffle split.

Since this process is repeated across all the three shuffle splits, $K \times n$ model training runs are conducted on different training/validation combinations. Every sample is trained $K \times (n - 1)$ times and validated K times. This will allow for less noisy estimations of a specific model validation performance.

For each one of the $K \times n$ train/validation combinations, we train the deep learning module on all combinations of the hyperparameters (5.2).

For each fold-based training/validation combination, there are $16 \times 2 = 32$ model hyperparameter combinations. Given that there are $K \times n$ fold-based combinations, the total amount of training runs amounts to $K \times n \times 32$. This comprehensive grid enables a non-biased selection of the best-performing model architecture and hyperparameter configuration.
To report the train and validation performances, we save the MSE and MAEs of training and validation set. In case feature maps constraints are also involved in training, we also consider the MSE of the maps for training and validation.
We launch the training for each run $R_{i,j,c}$ where fold j in shuffle split SS_i is used as the validation

set and the remaining three folds as the training set with a hyperparameter combination denoted as c. For each run, the model is trained for n epochs on the training set and the validation performance is monitored. The checkpoint with the lowest validation MAE is selected . That score represents the validation performance of the model with hyperparameter combination C on $F_{i,j}$.

The model training, validation and selection procedure is summarized in the Algorithm 5.1.

Algorithm 5.1 Selection of the best hyperparameter configuration based on validation performance using repeated CV

```
for $c \in C$ do
    // Loop over all configurations
    for $i \in [1, K]$ do
        // Loop over shuffle splits
        for $j \in [1, n]$ do
            // Loop over folds
            Select Fold $F_{i,j}$ as validation and the other three folds combined for training
            Compute the per-channel normalization weights for LIO and Emovi samples
              separately
            Construct a composed data sampler that combines samples from both sets
            Train the model for 50 epochs and select the optimal validation epoch
            $\mathrm{MAE}_{i,j,c} \leftarrow$ validation performance of run $R_{i,j,c}$
        end for
    end for
    $\mathrm{MAE}(c) \leftarrow \frac{1}{3\times4} \sum_{i=1}^{3} \sum_{j=1}^{4} \mathrm{MAE}_{i,j,c}$;
end for
$c^* \leftarrow \arg\min_{c \in C} \mathrm{MAE}(c)$;
```

Final training

After choosing the optimal final hyperparameter combination, we train the model on all the train set with the same training conditions (learning rate, optimizer, etc.) and we save the model checkpoint after training. Since the hyperparameter selection was performed using a repeated K-fold, the hyperparameters should be optimal for the final training on the complete train set. We trained two final models, one using the standard dataset ($\mathrm{Model}_{\mathrm{stand}}$) and the other one on the inpainted dataset (Model$_{\text{inp}}$). We trained the two models on all the trials and participants of the train set for the same number of epochs (50). For each trial, the last 20% of the sequence is set as validation to select the optimal validation performance while the first 80% of the set are left for training. We made sure to include a temporal train/validation split to have at least 5 whole gait cycles on the validation set. If we opted to randomly split each sequence, we might end up with some sequences having more predominant gait cycle phases in one of the sets. Each model is trained and tested in the same type of dataset (standard or inpainted).

Final testing

After training the final model, we test it on the holdout test set. Afterwards, we normalize the test set gait cycles (ground-truth and predictions) and compute the mean NGCs: the cycles are normalized according to the procedure described in Subsection 1.4.2. For each trial tr_i we get n_{t_i} normalized cycles of length 101 each (0 to 100). Those ground-truth NGCs are denoted $NGC_{tr,j}$ where tr is a specific trial and j is the index of the trial (for example 10^{th} trial). The same is done for the predicted NGCs denoted $\widehat{NGC_{tr,j}}$.

Next, we compute the key gait parameters (explained in SubSection 1.4.2): flexion heel strike angle at 0%, the peak flexion angle during loading response, the peak extension angle in terminal stance and the peak flexion angle in swing.

We also save the intermediate feature maps (the ones used as constraints) plots of the tested images. As discussed in the literature review, several existing methods estimate geometric angles based on 2D joint coordinates. We chose a benchmark model for HPE: HRNet that was trained on the COCO dataset on a specific configuration (pose_hrnet_w48_384x288) of 17 body joints, 48 channels width and an input resolution of 384×288. We directly used the estimations of this pretrained model without any fine-tuning or retraining and then extracted the lower body joints coordinates. Since HRNet typically detects the person via a YOLO-based person detector then estimates the pose from that full-body BB, we fed the entire image to the pipeline. Finally, we computed the joint angles from those coordinates similarly to the method used in Section 3.2.

5.3 Training and testing other SOTA approaches

As discussed in the literature review, several existing methods estimate geometric angles based on 2D joint coordinates. We chose a benchmark model for HPE: HRNet that was trained on the coco dataset on a specific configuration (pose_hrnet_w48_384x288) of 17 body joints, 48 channels width and an input resolution of 384×288. We directly used the estimations of this pretrained model without any fine-tuning or retraining and then extracted the lower body joints coordinates. Since HRNet typically detects the person via a YOLO-based person detector then estimates the pose from that full-body BB, we fed the entire image to the pipeline. Finally, we computed the joint angles from those coordinates similarly to the method used in Section 3.2.

We also trained HRNet from scratch on our dataset using a slightly different configuration that matches the one we used in our model. Since HRNet can be trained on any defined input image resolution, we set it to 224×224. We also set the amount of output joints to 3 (Knee, ankle and femoral head) and trained the pose detector directly on the lower body BB detections from our predefined detector (5.1.3). The data augmentations of the model were also changed to match the same ones we set for our model. Then we similarly infer the flexion angle from the predicted joints coordinates.

Another approach we used is 6DRepNet that outputs Euler angles based on the prediction of a 6D orientation representation. Since the pretrained model of this pipeline concern the head pose estimation, we trained our model from scratch by just using the pretrained RepVGG backbone (ImageNet weights). Using the equations 3.3, 3.4 and the ground truth annotations of the tibial and femoral anatomical frames, we can compute the 3×3 matrix associated with each annotation. The model computes the 6D representation from the input image and then transforms it into a 3×3 rotation matrix using the equation 1.7. Then the model is trained by optimizing the geodesic loss of the predicted rotation matrix and the ground-truth matrix. Finally, the Euler sequence is extracted from the rotation matrix (3.5) where the flexion angle is the first angle of the sequence. This model is also trained using the same resolution, BB and augmentation settings as the HRNet we trained from scratch.

All the models that were trained from scratch used the same train set as described in Section 5.1.1.

Similarly to our main approach, we test these models on the same test set and we extract the NGCs.

5.4 Hyperparameter selection and pipeline evaluation

To determine whether a model is performant, a well-defined evaluation scheme must be established to assess its various aspects. First, we start by introducing useful terms for the evaluation process.

The reference data (ground-truth) θ used to evaluate the model used for evaluation is generated following the steps of Chapter 4. The predictions $\hat{\theta}$ correspond to the outputs produced by the trained model to evaluate. The primary used evaluation metrics are:

- The signed error:

$$E = \hat{\theta} - \theta \tag{5.7}$$

- The absolute error

$$E = \left|\hat{\theta} - \theta\right| \tag{5.8}$$

- The squared error:

$$E = (\hat{\theta} - \theta)^2 \tag{5.9}$$

The average of both the absolute error and the squared error on the test set are respectively the MAE and MSE.

We denote by NGC and $\widehat{NGC}$ respectively the the NGC and the normalized predicted gait cycle. If the NGCs are grouped per trial, we denote the jth NGC of the trial tr and the jth predicted NGC of the trial tr respectively by $NGC_{tr,j}$ and $\widehat{NGC_{tr,j}}$. The absolute error between these terms is the the jth NGC AE relative to trial tr:

$$AE^{NGC}_{tr,j}(t) = \left|NGC_{tr,j}(t) - \widehat{NGC_{tr,j}}(t)\right| \tag{5.10}$$

with t the index of the NGC (from 0 to 100).

If we group the NGCs of all the trails of the same participant P, the AE relative to the NGC of the participant P is:

$$AE_{P,j}^{NGC}(t) = \left| NGC_{P,j}(t) - \widehat{NGC_{P,j}}(t) \right| \tag{5.11}$$

The MAE NGC is the mean of all the AE NGCs grouped by trial (MAE_{tr}^{NGC}), by participant (MAE_{P}^{NGC}) or all trials included (MAE^{NGC}).

This section is divided into multiple subsections, each corresponding to a specific evaluation axis. The evaluation pipeline mainly includes several key components: evaluating the CV, testing the final model (trained in the entire training set) on both global (all trials mixed) and trial-wise (each trial individually) levels, assessing the impact of image inpainting and the feature maps constraints, and comparing the model with other SOTA approaches.

5.4.1 Model/hyperparameter selection

This step describes the evaluation process used during the hyperparameter tuning and model selection phase, based on repeated CV. First, the participant-wise data partitioning of the CV procedure is presented to ensure training folds, validation folds and the test set are evenly distributed.

During each CV iteration, the local model performance is assessed using the MSE and MAE computed on both training and validation sets in order to monitor convergence of the CV trainings and generalization throughout the training process.

To evaluate the impact of feature map constraints on the model learning, the corresponding loss components (the constraints loss terms) are tracked across training epochs.

The optimal hyperparameter configuration is selected based on the average validation performance across all cross-validation folds, using MAE as the primary selection criterion.

5.4.2 Model performance and flexion angle error analysis

This step outlines the evaluation process used to assess the model's global performance, the trial-wise behavior, and compare it to the other SOTA methods.

The final trained model (trained on the optimal hyperparameter combination from Subsection 5.4.1) is evaluated. Similarly to the model/hyperparameter selection evaluation, the model performance during the final training run is monitored using the MSE and MAE, computed on both the training and validation sets.

The model performance is then assessed on the holdout test set. The test results are all aggregated globally across all trials using MAE and MSE as the evaluation metrics. To identify potential model inconsistencies, prediction errors are monitored to detect outliers or abnormal patterns.

Model performance is compared against the results of several baselines: a pretrained HRNet, a retrained HRNet on our dataset, and a trained 6DRepNet on our dataset. The evaluation of all the models is done under the same conditions.

To statistically assess the performance difference between our proposed model (with constraints) and the other SOTA approaches, we apply the SPM1D framework. We compare the NGCs AE of our model ($AE^{NGC}[const]$) with the one relative to the trained HRNet (AE_HRNet^{NGC}) and to 6DRepNet (AE_6D^{NGC}) with 2 different tests.

The respective null hypothesis for the 2 tests are:

$$H_0 : MAE^{NGC}[const](t) - MAE\_6D^{NGC}(t) \leq 0 \tag{5.12}$$

and

$$H_0 : MAE^{NGC}[const](t) - MAE\_HRNet^{NGC}(t) \leq 0 \tag{5.13}$$

Similarly to the aggregated test set (all trials), we assess the test performance trial-wise and patient-wise. We also compute the number of gait cycles per trial (Subsection 13, Section 5.2). The $AE^{NGC}_{tr,j}$ curves are compared to a certain threshold m_0 using the SPM1D t-test under the null hypothesis H_0 assuming that the mean (on all cycles of the same trial) absolute error of the

NGC does not exceed a certain limit m_0:

$$H_0 : MAE_{tr}^{NGC}(t) \leq m_0 \tag{5.14}$$

Since we want a clinically valid model estimation, we set m_0 to a clinical standard threshold. On a similar basis, we do the SPM1D test on participant-level. The null hypothesis of the SPM1D test is:

$$H_0 : MAE_{P}^{NGC}(t) \leq m_0 \tag{5.15}$$

On trial-level, we also evaluate the model's ability to predict clinically relevant gait parameters (listed in Subsection 13, Section 5.2). For each trial, we compute the listed gait parameters from the predicted and ground-truth average NGCs. The differences in magnitude and phase (index in the gait cycle from 0 to 100) are calculated to assess prediction accuracy at the kinematic level. For a ground-truth gait parameter GP extracted from a certain trial NGC and its predicted counterpart $\widehat{GP}$, their indexes are denoted idx_{GP} and $idx_{\widehat{GP}}$ The magnitude and index deviations are respectively given by the equations:

$$E_{GP}^{idx} = idx_{\widehat{GP}} - idx_{GP} \tag{5.16}$$

$$E_{GP}^{idx} = \widehat{GP} - GP \tag{5.17}$$

5.4.3 Ablation Study: Inpainting and feature maps constraints

This step evaluates the impact of integrating image inpainting and FM constraints into the model training pipeline.

To assess the effect of image inpainting, four combinations of training and testing configurations are considered:

- Model trained on the standard dataset and tested on the standard dataset.
- Model trained on the standard dataset and tested on the inpainted dataset.
- Model trained on the inpainted dataset and tested on the standard dataset.

- Model trained on the inpainted dataset and tested on the inpainted dataset.

For each configuration, performance is evaluated on the test set using the AE as the primary metric. Additionally, intermediate feature maps (the feature maps selected for the constraints) are extracted for samples from each configuration as a qualitative evaluation of the learned representations under different training conditions.

To evaluate the impact of adding FM constraints to the training objective, we evaluate and compare the test AE of the two models trained with and without FM constraints (Model$_{\text{stand}}$ and Model$_{\text{inp}}$).
To statistically assess whether the use of constraints leads to a significant difference in performance, a SPM1D t-test is conducted for all the NGCs of all the trials aggregated. The statistical hypothesis test evaluates whether the mean of the AE NGCs relative to the constraint-based model is significantly lower than the mean AE^{NGC} of the constraint-free model:

$$H_0 : MAE^{NGC}[const](t) - MAE^{NGC}(t) \leq 0 \tag{5.18}$$

Conclusion

This chapter presented the complete methodological steps for the estimation of knee flexion angles from monocular RGB images. We detailed the processing of the raw data, including the use of data augmentation, image inpainting and CV-based data splitting. The deep learning pipeline was outlined, with focus on the training procedure, the integration of feature map constraints, the strategy for hyperparameter tuning through repeated CV and the the final model testing. The evaluation protocol with the appropriate evaluation metrics was also defined highlighting both global and trial-level assessments, the comparison with other SOTA approaches, the feature maps and image inpainting constraints, and the corresponding SPM tests.

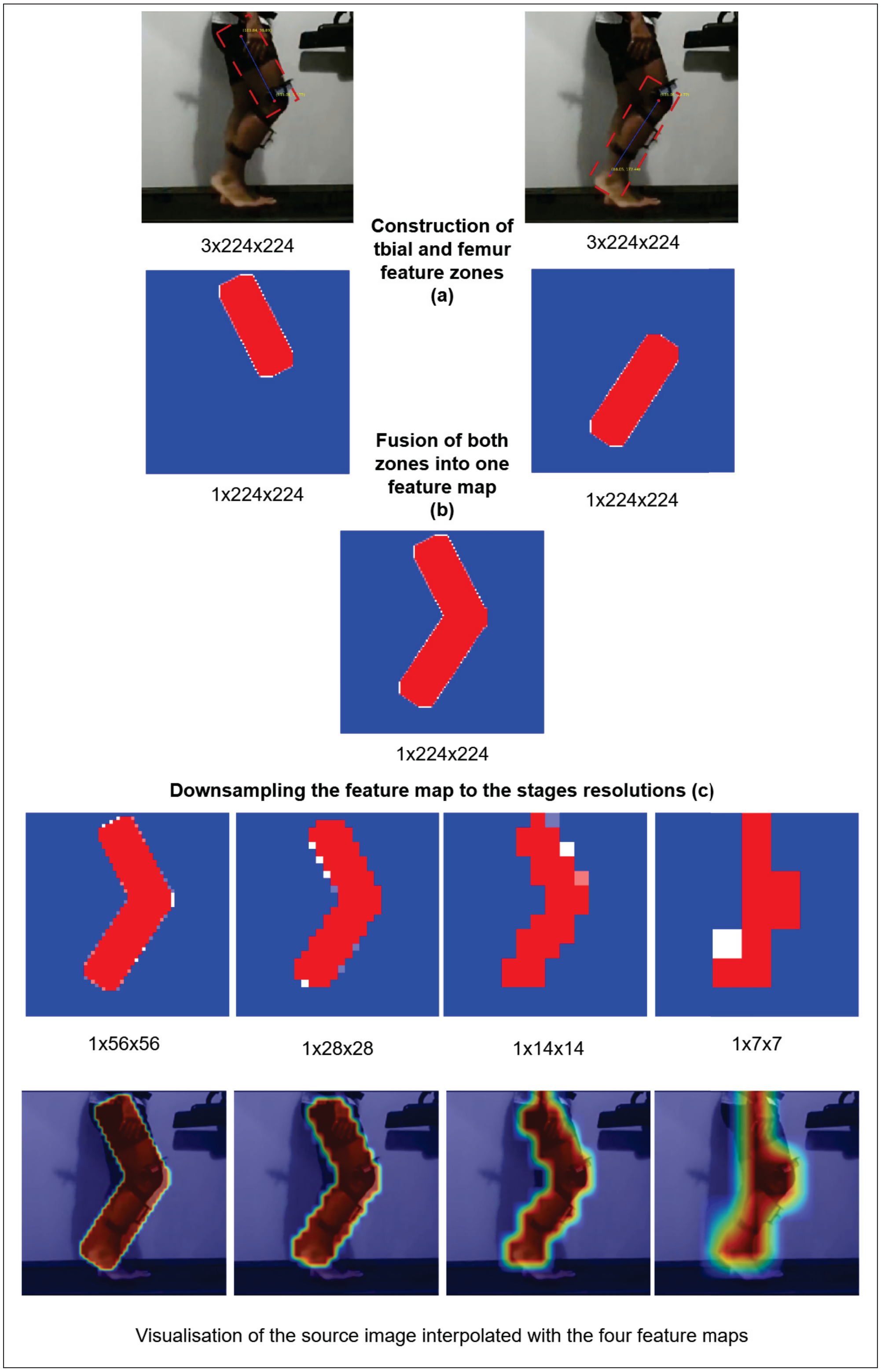

3x224x224
Construction of tbial and femur feature zones (a)
3x224x224
1x224x224
Fusion of both zones into one feature map (b)
1x224x224
1x224x224
Downsampling the feature map to the stages resolutions (c)
1x56x56
1x28x28
1x14x14
1x7x7
Visualisation of the source image interpolated with the four feature maps

Figure 5.8 Pseudo-label Feature maps creation

CHAPTER 6

EXPERIMENTAL RESULTS AND DISCUSSION

Introduction

In this chapter, we summarize the CV results and evaluate the final trained model on the test set. First, we analyze and interpret the results of the repeated CVs and the hyperparameter selection results. Then, a first high-level analysis will be conducted to assess the models test performance on both the standard and inpainted dataset. We also present a comparative study of the performance with two other benchmarks. A second study will include a more in-depth error analysis on trial-level. Afterwards, we highlight the benefit of the inpainting task and its impact on the performance. Finally, we do a comparison with other SOTA methods.

6.1 Model/hyperparameter selection

Results

As mentioned in Section 5.1.1, we use the repeated CV split for the dataset. The holdout test contains $n_{test} = 5$ participants with exactly 2 trials each. The remaining $n_{train} = 25$ participants are split into $n = 4$ folds with $K = 3$ different splits.

The Table 6.1 shows the adopted participants repartition of the dataset.

Table 6.1 CV patient splits

Test	**Shuffle split 1**				**Shuffle split 2**				**Shuffle split 3**			
	Fold1	Fold2	Fold3	Fold4	Fold1	Fold2	Fold3	Fold4	Fold1	Fold2	Fold3	Fold4
P05 P06 P15E P20 P24	P0E P11 P27 P31E P16E P18 P21	P02E P10 P17 P04E P26 P28	P29E P13 P22 P07E P12 P23	P03 P14 P30E P08E P19 P25	P10 P19 P04E P31E P02E P25 P26	P27 P14 P16E P30E P11 P13	P0E P12 P28 P08E P18 P22	P07E P17 P21 P29E P23 P03	P11 P0E P19 P08E P28 P03	P27 P29E P17 P22 P26 P13	P10 P25 P21 P12 P30E P02E P04E	P14 P16E P31E P18 P23 P07E

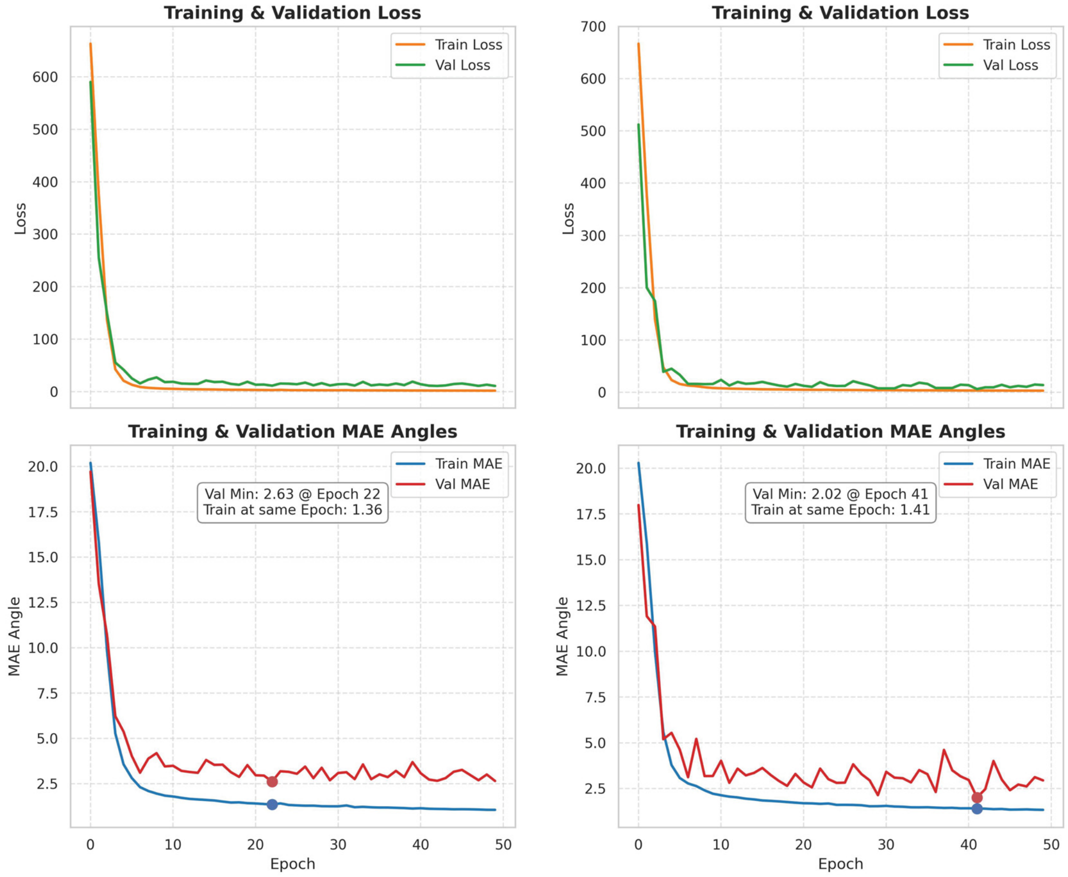
Since the dataset was collected in two distinct environments—the LIO laboratory and the Emovi office—using the same acquisition setup (but with differences in background and camera distance), we distinguish between participants recorded at the Emovi office by appending the letter "E" to their IDs.

First step is to training and validation curves of the CV runs. Due to the high number (768) of cross-validation runs and the observation that most runs exhibit similar convergence patterns and only some differences in the reached error value during convergence mainly due to the differences in participants distribution, we limit our detailed analysis to a subset of representative runs. With this method, we can ensure clarity and interpretation of the results while still capturing the general convergence behavior of the model across the different CV runs. We select the instance relative to the third CV run of the second shuffle split (Table 6.1) on the standard dataset $R_{2,3,c_{rep1234}}$ for evaluation. This corresponds to the 16^{th} line and 7^{th} column of the annex IV-1. $c_{rep1234}$ is the hyperparameter combination relative to a RePVGG backbone with feature maps constraints applied to the stages 1, 2, 3 and 4. We compare the two convergence curves of the cost (MSE) and MAE values and the train and validation datasets with their counterparts from $R_{2,3,c_{rep}}$ (no constraints applied) which corresponds to the 1^{st} line and 7^{th} column of the same annex IV-1 The plots are shown in Figure 6.1. The blue and red dots on the MAE curves represent respectively the train and validation MAEs when the MAE loss is minimal.

We also plot in Figure 6.2 the train and validation feature maps curves of the same configuration $c_{rep1234}$ to confirm if the constraints curves converge or not.

We report the validation MAE of the repeated CV results for the experiments on the standard (uninpainted) dataset and for the inpainted dataset in Appendix IV.
The best average validation scores for both models (standard and inpaint) are listed on this Table 6.2 with the corresponding hyperparameter choice. The values represent the average of all the validations MAEs on all the splits for those hyperparameter combinations.

The average validation results of the best standard and inpainted validation errors are highlighted in bold respectively on the annexes IV-2 and IV-4.



a) No constraints applied to feature maps

b) Constraints applied on features maps of stages 1, 2, 3 and 4

Figure 6.1 Train and validation convergence plots on the model trained without constraints and the model trained with constraints: MSE and MAE (°)

Discussion

According to the Figure 6.1, the training curves in blue show consistent and gradual convergence, indicating proper learning. The validation curves in red demonstrate that there is a reduction of overfit when constraints are applied to the training (less visible for the MSE cost function because of the scaling). The model reaches a better validation MAE after fewer epochs. The feature maps help the model generalize better on new participants (not seen in training) data and reduce the overfit.

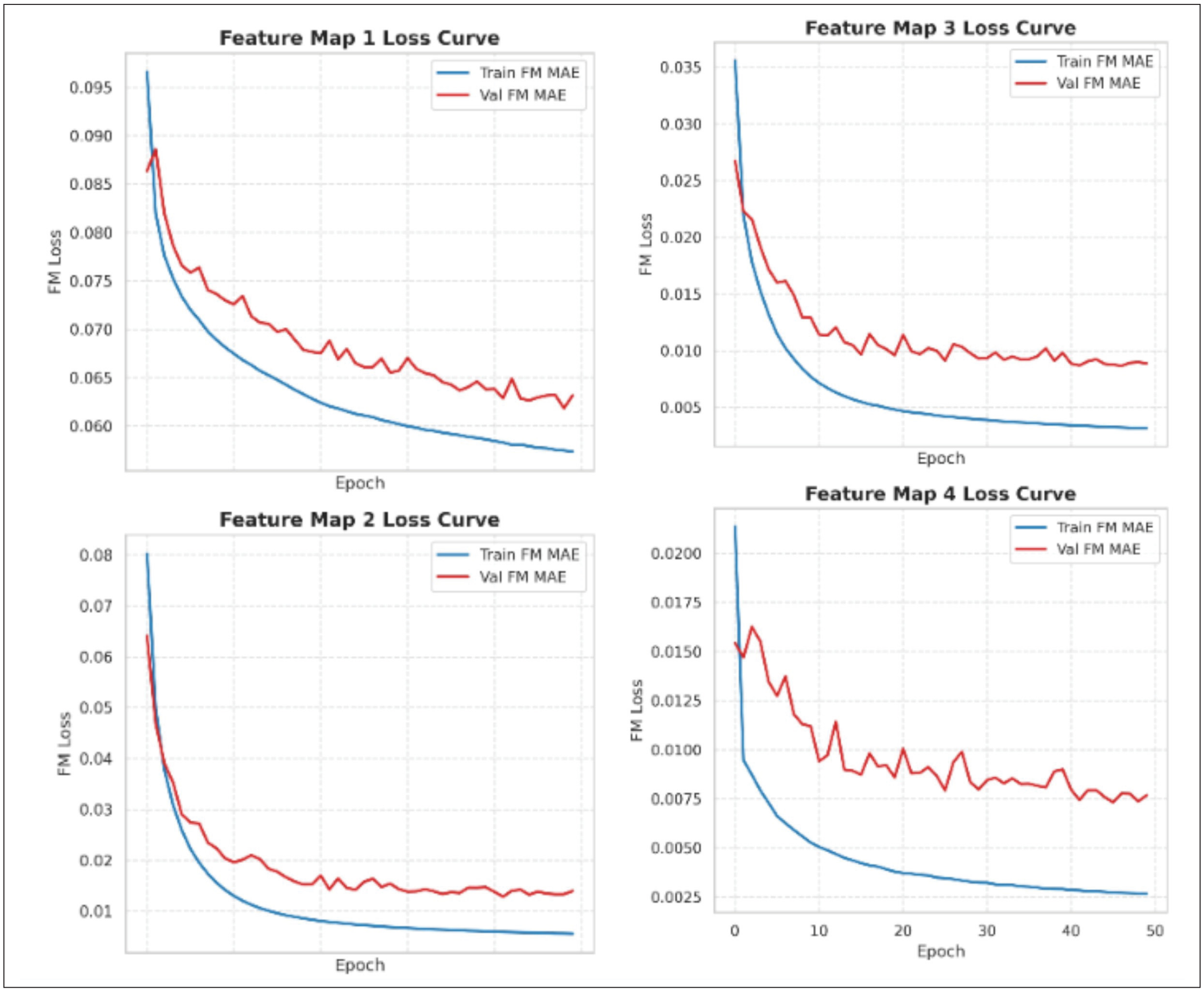


Figure 6.2   Feature maps constraints training and validation curves

The Figure 6.1b shows that the feature maps constraints but more advanced stages of the network tend to show a bit more overfitting than the previous ones. This can be explained by the fact that backbones extract more advanced and specific features on later stages of the network.

The Table 6.2 shows that for both standard and inpainted datasets, the CV results perform the best with a ResNet backbone and a specific combination of feature map constraints weights. This concludes that ResNet has a better capacity to learn from the data than the RePVGG backbone. This assumption is also supported by the overall results of the annexes IV-1, IV-3, IV-2 and IV-4: the ResNet validation results on the different trials are overall better than RepVGG. The optimal

Table 6.2 Models average MAE performance on standard and inpainted datasets

	Standard dataset	**Inpainted dataset**
Model name	$\text{Model}_{\text{stand}}$	$\text{Model}_{\text{inp}}$
Backbone	ResNet	ResNet
λ_1	1000.0	0
λ_2	0	1000.0
λ_3	0	0
λ_4	1000.0	1000.0
Validation MAE(°)	2.21	2.9

hyperparameter combination differs according to the type of dataset and the annexes also show that for each shuffle (from 1 to 3), there might be a difference of feature maps constraints coefficients choice. This is why we opted for a repeated CV: it tends to reduce the bias based on the grouped participants per fold. The more shuffle splits we have, the less biased the results are. A higher number (than 3) of splits could be used but at the expense of longer computation time.

6.2 Model performance and flexion angle error analysis

In this section we report and then interpret several results related to the model's performance:

The model global performance : the error analysis of the flexion angle including all trials and participants.

Comparison with SOTA : compare our model's test performance with the SOTA models (HRNet + 6DRepNet) global performance.

The model performance per trial/participant : the error analysis of the flexion angle per trial including key gait parameters.

Results

We plot the training and validation curves (MSE and MAE) of the final $\text{Model}_{\text{inp}}$ for reference. The Figure 6.3 shows the convergence plots.

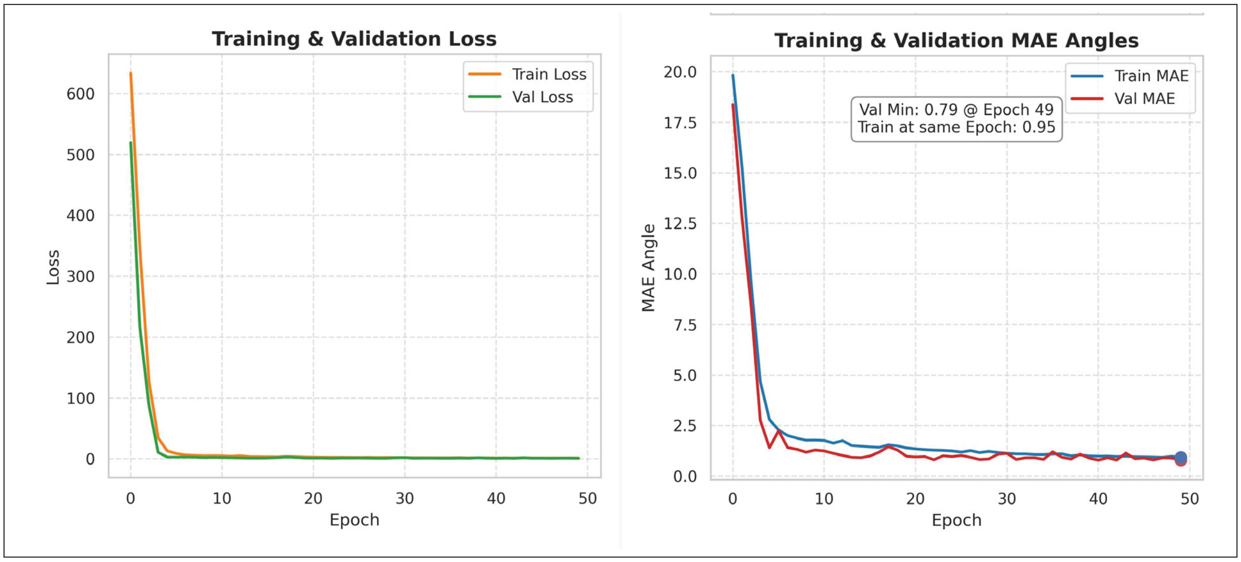


Figure 6.3 Train and validation MSE and MAE plots of he final $\text{Model}_{\text{inp}}$

The final test results on the test set are presented in the Table 6.3:

Table 6.3 Selected hyperparameters and average MAE for standard and inpainted datasets

	$\textbf{Model}_{\textbf{stand}}$	$\textbf{Model}_{\textbf{inp}}$
Test MAE(°)	3.12 ± 1.86	2.40 ± 1.92

$\text{Model}_{\text{inp}}$ has a better performance (lower MAE) than $\text{Model}_{\text{stand}}$.

We show the distribution of the ground-truth, predictions, errors and AEs of all the predictions of all the merged test trials in the Figure 6.4. We also produce the Bland-Altman (Martin Bland & Altman, 1986) plots to determine where along the range of angles your model tends to deviate - underestimate or overestimate) more from the ground truth. The X-axis contains the average ($x = \frac{\widehat{\theta}+\theta}{2}$) of the two measurements (ground-truth and predicted angles) and the Y-axis is the difference between the two measurements for each measurement ($y = \widehat{\theta} - \theta$).

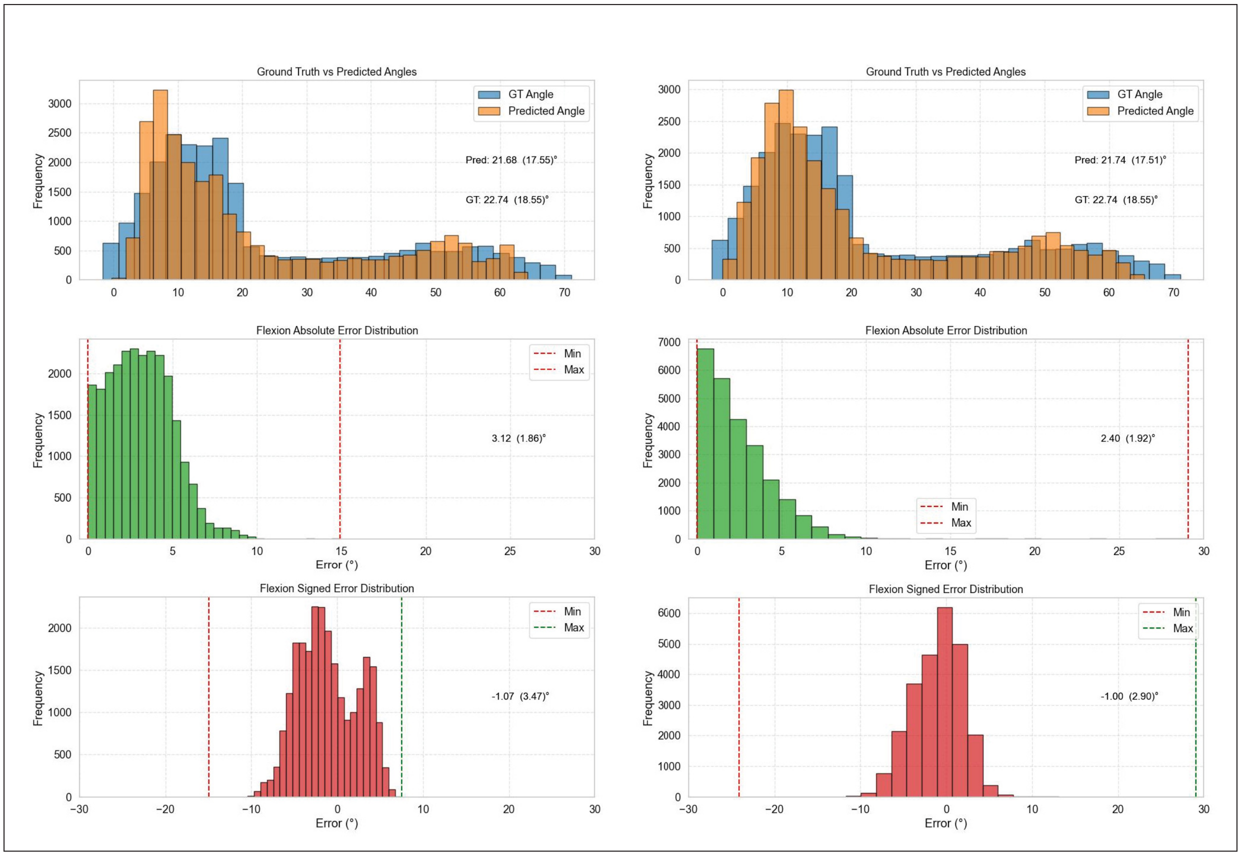
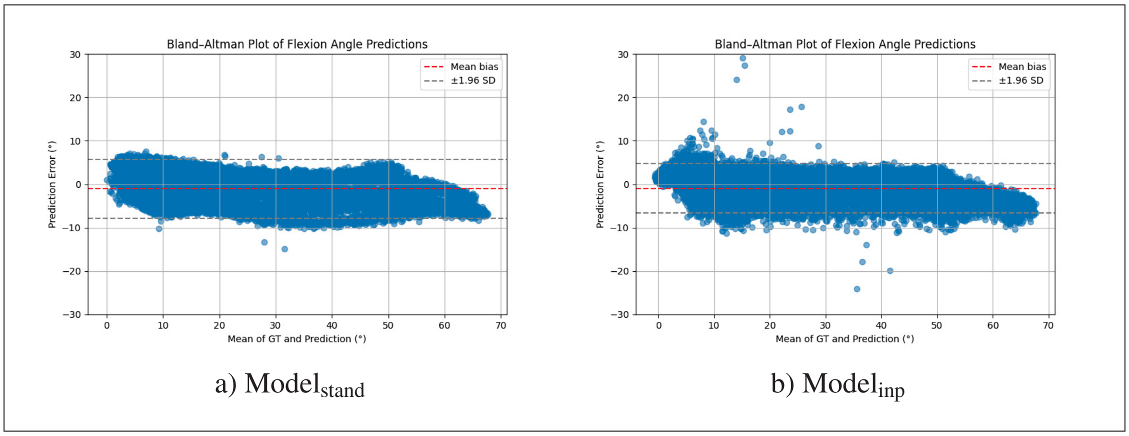


Figure 6.4 Ground-truth, predictions, signed errors and AEs of the flexion angle distributions for Model$_{\text{stand}}$ and Model$_{\text{inp}}$



a) Model$_{\text{stand}}$

b) Model$_{\text{inp}}$

Figure 6.5 Bland-Altman plots relative to Model$_{\text{stand}}$ and Model$_{\text{inp}}$

To investigate why Model$_{\text{inp}}$ has more extreme error values (outliers), we refer to some of these images with a high AE. An example of an image before and after inpainting from a trial of participant P24 is shown in the Figure 6.6:

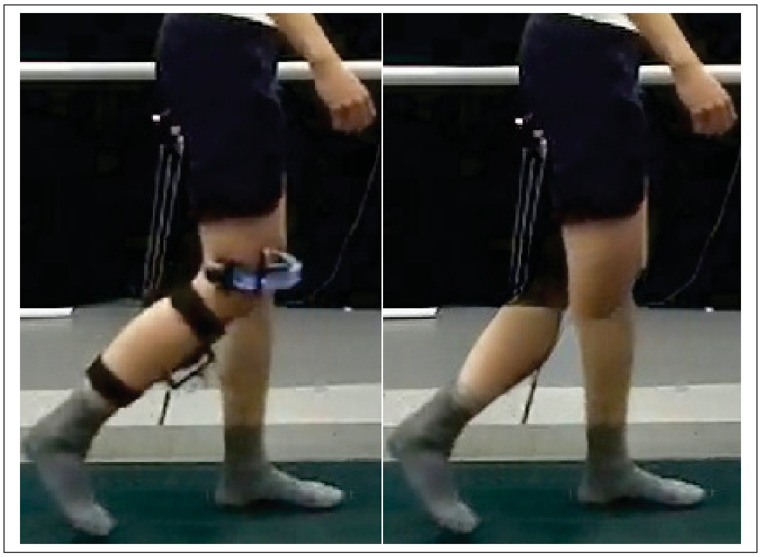


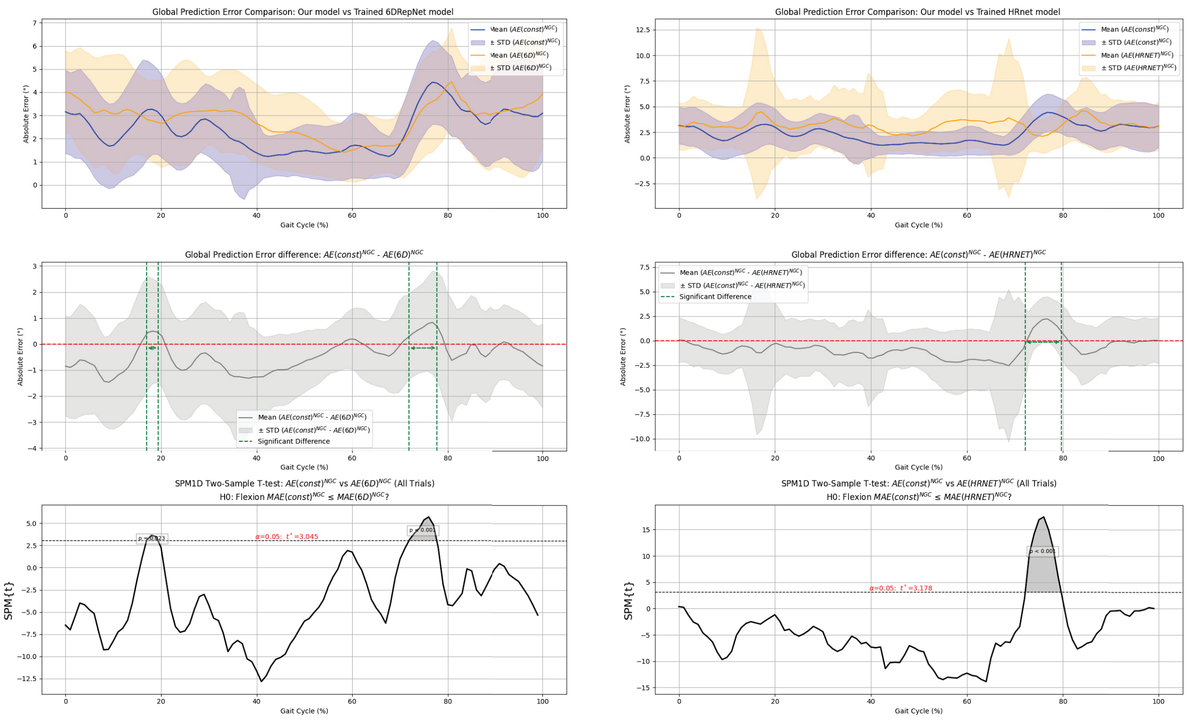
Figure 6.6   Example of a bad image inpainting leading to a high angle prediction error

Next, we compare our model's performance with the other SOTA approaches.
We report the test MAEs of all the models (pretrained HRNet, Retrained HRNet, 6DRepnet, our model without constraints and our model with constraints) on the test set in the table 6.4.

Table 6.4   Comparison of model performance

Model/Pipeline	Pretrained HRNet	HRNet trained from scratch	6DRepNet	Our model (no constraints)	Our model (with constraints)
MAE (°)	4.39 (5.53)	3.36 (3.69)	2.90 (2.12)	2.66 (2.05)	**2.40 (1.92)**

Our model vs HRNet & 6DRepNet: Our model (no constraints) shows a lower MAE than the trained 6DRepNet (17.24% improvement) and the trained HRNet (40.67% improvement). To test the significance of these assumptions over the gait cycles, we do a the SPM1D-t test. The results of both tests are shown in the Figure 6.7.



a) Our model (with constraints) vs 6DRepNet

b) Our model (with constraints) vs HRNet

Figure 6.7 SPM1D-t: Does our model have a lower MAE than HRNet and 6DRepNet?

In the remaining part if this paragraph, we present the results per trial/participant.

The table 6.5 shows the test results per trial for both **Model$_{\textbf{stand}}$** and **Model$_{\textbf{inp}}$** on their respective datasets. The angle results are reported under the format mean(standard deviation) in degrees. The plots for the ground-truth flexion angle values and the predictions for both models (**Model$_{\textbf{inp}}$** and **Model$_{\textbf{stand}}$**) are in Appendix I. Each trial has a plot containing the ground-truth angles distribution, the predicted angles distribution and the model performance errors (signed and absolute) distributions on the designed dataset. The plots (for Model$_{\text{stand}}$ and Model$_{\text{inp}}$) for all the trials merged together (the whole test set) correspond to the plots on Figure 6.4.

Each trial has a different number of gait cycles depending on the trial length after data cleaning and the participant walking speed. In the Table 6.6, we show the number of NGCs per test trial.

Table 6.5 Mean Errors (°) and MAEs (°) for **Model**$_{\textbf{stand}}$ and **Model**$_{\textbf{inp}}$ across Participants and Trials

Participant	**Trial**	**Model**$_{\textbf{stand}}$		**Model**$_{\textbf{inp}}$	
		Test mean error	**Test MAE**	**Test mean error**	**Test MAE**
P05	001	1.59 (2.68)	2.70 (1.55)	0.55 (1.65)	1.45 (0.95)
	002	1.69 (2.51)	2.54 (1.64)	0.45 (1.62)	1.35 (0.99)
P06	111	-4.36 (2.20)	4.36 (2.20)	-2.99 (2.89)	3.35 (2.47)
	112	-4.34 (1.93)	4.34 (1.93)	-2.54 (2.62)	2.95 (2.14)
P15E	000	2.48 (1.93)	2.88 (1.25)	0.69 (1.96)	1.79 (1.06)
	001	2.61 (2.15)	3.09 (1.37)	1.25 (3.16)	2.77 (1.79)
P20	000	-4.23 (1.49)	4.24 (1.47)	-2.12 (2.49)	2.60 (1.98)
	001	-4.29 (1.56)	4.31 (1.51)	-2.83 (2.66)	3.27 (2.10)
P24	001	-1.52 (1.35)	1.68 (1.14)	-1.79 (2.63)	2.54 (1.91)
	002	-1.16 (1.30)	1.41 (1.30)	-0.70 (2.56)	2.03 (1.72)

The following tests results are based on the predictions of **Model**$_{\textbf{inp}}$ on the inpainted dataset. The plots of the mean GT and predicted NGCs per trial and per patient are in Appendix II. The appendix also contains the mean of the NGCs errors $E^{NGC}_{tr,j} = \widehat{NGC_{tr,j}} - NGC_{tr,j}$.
We set $m_0 = 2°$ from Equation 5.14 as it is the benchmark difference in estimation compared to MoCap systems. The obtained SPM1D plots are in the Appendix III. We show an example of participant P20 in the Figure 6.8 and the SPM1D applied to all the 76 trials of that participant.

The first plot of Figure 6.8 shows the mean and standard deviation of the AE values relative to all gait cycles at that time t with $t \in [0, 100]$. The second plot is the SPM1D statistic with the gray areas the intervals where we fail to reject the null hypothesis $H_0 : MAE < m_0 = 2°$.
The values of errors of the key gait parameters (predicted gait parameters - ground-truth gait parameters) are reported in the Table 6.7. The higher the gray area the higher the confidence of

Table 6.6 Overview of normalized nait Cycles per Participant and Trial

Patient ID	Trial ID	Number of gait cycles
P05	001	33
	002	34
P06	111	35
	112	35
P15E	000	41
	001	40
P20	000	38
	001	38
P24	001	41
	002	41

rejecting H_0. The green dashed lines and arrows project the same regions on the first plot. The blue dashed lines show the moment (indexes) in the mean ground-truth NGC when certain key gait cycle phases occur.

Table 6.7 Knee flexion angle estimation errors at key gait events (°)

Patient	Trial	Flexion heel-strike	Max flexion during loading	Max extension during stance	Max flexion
05	001	-0.15	0.71	1.56	2.32
05	002	-0.22	0.97	1.23	2.12
06	111	-5.23	-4.39	0.56	-4.08
06	112	-4.37	-5.71	0.82	-3.68
15E	000	2.42	0.5	1.36	-3.42
15E	001	2.56	0.83	2.67	-3.87
20	000	-2.57	-3.35	0.01	-4.95
20	001	-2.79	-3.66	-1.41	-5.14
24	001	3.60	-2.90	-1.19	-5.82
24	002	4.56	-1.87	-0.91	-4.68

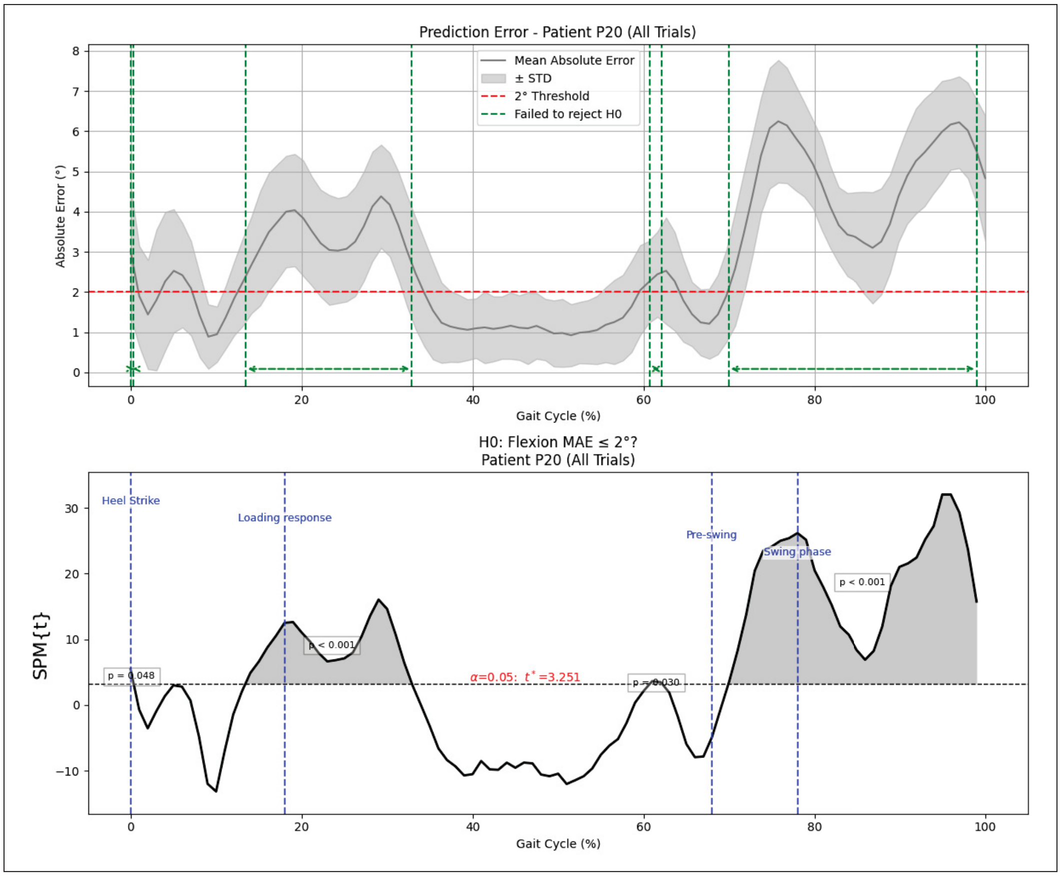


Figure 6.8 Mean, standard deviation and SPM1D-t plots for participant P20

More details about the index deviations of each gait parameter are in the figures of Appendix II.

Discussion

The plots in Figure 6.3 exhibit smooth and stable training and validation convergence plots. It is expected to get almost no overfit in the final training because the train and validation sets contain the knee images of the same person with the same clothes and the same background doing the same movement. The reason behind setting the validation set is not to assess the validation performance but to set a stopping condition and make sure the model does not diverge.

In general, the model results (Table 6.3 and Figure 6.4) show clinically largely acceptable flexion angles below 5 degrees of error relative to marker-based Motion Capture systems (McGinley *et al.*, 2009).The predicted angles histograms (Figure 6.4) and more clearly the Bland-Altman plots (Figure 6.5) show that for both models there is more tendency to underestimate extreme high flexion angles (higher than 65 °) and overestimate extreme low angles (lower than 5 °). Moreover, the error values are predominantly negative: the flexion angle is overall underestimated (negative mean error value) with a mean error value of -0.96° for $\text{Model}_{\text{stand}}$ and -1.07° $\text{Model}_{\text{inp}}$. Although the model trained on an inpainted dataset shows a better overall MAE (Table 6.3 and Figure 6.4), it can yield more extreme error values (Figure 6.4 and Figure 6.5). This is mostly due to some inpainting inconsistencies where the removal of the KneeKG considerably corrupts some parts of the leg which makes the feature extraction for the model more difficult. Since the model was trained using feature constraints that emphasize specific regions of the leg, suboptimal inpainting that distorts the knee structure can lead to inaccurate predictions. The Figure 6.6 shows an example of that inconsistency.

Next, we interpret the test results of HRNet and 6DRepNet from Table 6.4:

- **6DRepNet**: Although the performance of 6DRepNet seems to be comparable to our model, it yields a significantly higher degree of overfitting. This can be explained by the fact that 6DRepNet estimates a full 3D transformation (from the AF to anatomical AT frame) while the actual movement in our trials is predominantly in the sagittal plane. Incorporating information from the frontal and transverse planes adds unnecessary complications to the estimation process of the flexion angle (which can be reliably inferred using just the sagittal plane information) and reduces its efficiency. Estimating the additional abduction and rotation angles such as abduction or rotation angles from a lateral view is not only hard but also forces the model to optimize an objective function conditioned by those three angles. Hempel *et al.* (2022) were able to apply the 6D pose estimation method in a different for head pose estimation where the images were captured from different viewpoints providing a variety of Euler angles variations. For our case, focusing on only the visually accessible sagittal plane features is the more robust and accurate approach to estimate the flexion angle.
- **HRNet**: The HRNet pretrained model shows a high mean error and variance. Since it was trained on COCO dataset, the image domain change and the change in the annotations nature (from manually annotated to clinically supervised keypoint annotation) make the

keypoint's estimations inaccurate resulting in a high angular AE. While retraining HRNet on our dataset seems to significantly reduce the problem of the data domain difference compared to the results from the pretrained instance, the predictions errors still stack up to obtain a higher angular error with a high variance. HRNet is a very deep convolutional network that performed well with large-scale and diverse datasets like MPII and COCO pose datasets. Our dataset has a simpler structure and is of a much smaller scale; which can explain that HRNet might to some extent overfit on the data.

The interpretations of the SPM plots from Figure 6.7 are similar to the ones from the Figure 6.11. We only reject the null hypothesis for a small interval on less than 10% of the gait cycle near the start of the swing phase (end of the pre-swing and start of toe-off phases). Therefore there is a probability of 95% that the model trained with constraints performs better than the trained 6DRepNet and HRNet on at least 90% of the gait cycle.

The following discussion is an in-depth interpretation for the results of each trial and patient individually.
The Table 6.5 shows that for all trials except the trials of participant P24 (8/10 trials), the MAE of $\text{Model}_{\text{inp}}$ is inferior to the MAE of $\text{Model}_{\text{stand}}$. This confirms the hypothesis of $\text{Model}_{\text{inp}}$ having a better performance than $\text{Model}_{\text{stand}}$. The Figure 6.6 shows an inpainted image from the same participant P24 showing inpainting inconsistencies that caused some parts of the Knee to be removed which can explain the inpainting causing the performance to be worse for that specific participant.
According to Table 6.5, we notice that for some participants (P05, P15E) there is an overestimation of the angles (the signed error mean is positive while for the others (P06, P20, 924) the angles are underestimated (negative signed error mean). This observation is valid for both models results. The overall error of the whole test set is predominantly negative: underestimated angles. For the participant P20 (Table 6.8), we can approximately interpret that the MAE has a 95% chance of being superior to $m_0 = 2°$ just before and during the start of the loading response and at the pre-swing and swing phase. We summarize our observations and interpretations in the Table 6.8 for the five test patients based on the SPM1D-t test results from Appendix III.

Table 6.8 Summary of participant-wise mean gait cycle statistical significance

Patient ID	Decision
P05	H_0 is accepted for mostly all the gait cycle with the exception of some minor parts at the start of the swing phase.
P06	We reject H_0 for approximately half the gait cycle: from heel strike to the onset of loading response, throughout the loading response to mid-stance and from midway through pre-swing to the end of the swing phase.
P15E	There is a tendency to reject H_0 around transition times in the gait cycle: the start of the cycle (heel strike), before the onset of pre-swing and during intervals near the initiation of the swing phase.
P20	We reject H_0 near the loading response and during the pre-swing and swing phases.
P24	We accept H_0 for most of the stance phase and by the start and end of the swing phase.

In general, there are no specific regions where H_0 is more likely to be rejected. The interpretation heavily depend from the participant and from the choice of m_0. However we notice that the null hypothesis is mostly accepted at the second part of the stance phase mostly between the limits of 35% and 65% of the NGCs which represent mostly the terminal stance and the pre-swing phases yielding lower flexion angle values.

6.3 Ablation Study: Inpainting and feature maps constraints

Results

First, we present the results relative to the image inpainting. To showcase its importance, we did a cross-dataset testing: Each trained model is tested on both datasets to assess the generalization capacity. For a quantitative evaluation, the MAE results are shown in Table 6.9 below where the columns present the used train dataset and the rows present the test one.

Table 6.9 Cross-dataset performance: MAE of the model trained and tested on standard/inpainted dataset(°)

		Train	
		Standard database	**Inpaint database**
Test	**Standard database**	3.12 (1.86)	3.44 (2.70)
	Inpaint database	16.21 (16.00)	2.40 (1.91)

The green-highlighted values correspond to each model evaluated on the same type of dataset it was trained on, representing ideal conditions. The cell highlighted in orange is the performance of the model trained on the inpainted dataset and tested on the standard dataset. The red error value is the performance of the model trained on the standard dataset when tested on inpainted images.

For a qualitative evaluation, we plot some feature maps extracted from the intermediate stages. Since $\textbf{Model}_{\textbf{stand}}$ was trained on ResNet with constraints applied to the intermediate feature maps of stages 1 and 4, we focus on these layers for analysis. The corresponding feature maps from two versions of the same test image are plotted: the original (standard) image and its inpainted counterpart.

The Figure 6.9 shows the backbone (ResNet) stages 1 and 4 output feature maps of the same image: the first row for the standard image (with KneeKG) and the second one for the inpainted image (KneeKG removed).

Similarly, we plot the feature maps of stages 2 and 4 (used for $\textbf{Model}_{\textbf{inp}}$ training) on the same image twice: once on the standard image and on the inpainted one. We plot the feature maps in the Figure 6.10:

The second part of the results is relative to the impact from the added feature maps constraints. We present the test MAEs in the Table 6.10 for the standard and inpainted datasets. We trained $\textbf{Model}_{\textbf{stand}}$ on the standard dataset and tested it on the standard test set under two settings: without the feature maps constraints and with the feature maps constraints (first column **Standard**). We did the same process for $\textbf{Model}_{\textbf{inp}}$ (second column **Inpaint**).

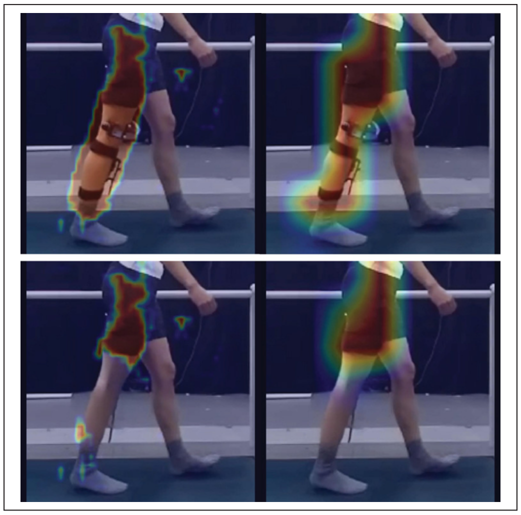


Figure 6.9 ResNet Feature maps at stages 1 and 4 from $\mathbf{Model_{stand}}$ comparison: standard images and inpainted image

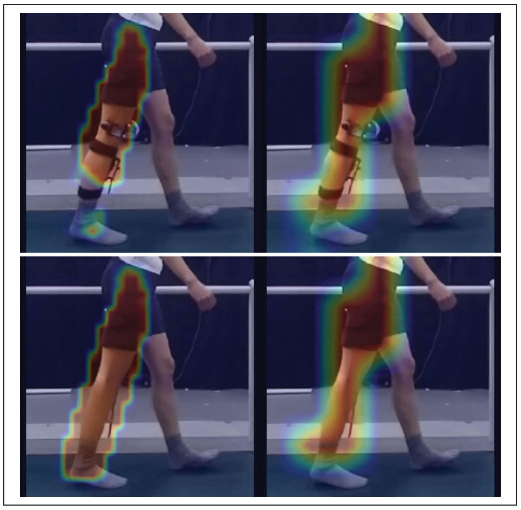


Figure 6.10 ResNet Feature maps at stages 2 and 4 from $\mathbf{Model_{inp}}$ comparison: standard images and inpainted images

There is an improvement on the test performance when the feature maps constraints are added to the cost function. In fact for both dataset settings (standard and inpainted), there is a reduction in the test MAE and the standard deviation.

To test if this improvement is statistically significant, we conduct the SPM1D-t test (Equation 5.18

Table 6.10 MAE (°) test performance under different settings. Values are in the format: mean (standard deviation)

Standard		Inpaint	
No Constraint	Constraint	No Constraint	Constraint
3.68 (2.24)	**3.12 (1.86)**	2.66 (2.05)	**2.4 (1.91)**

). The applied SPM1D-t statistical test on all the NGCs AEs is representative for all the test data. The results are plotted in the Figure 6.11:

The first plot contains the mean and standard deviation of the AEs of NGCs for all the trials for both models. The second plot is the mean and standard deviation of their difference $AE^{NGC}_{tr,j}[const] - AE^{NGC}_{tr,j}$. The third plot is the statistic curve: the gray region is where we reject the null hypothesis H_0 with confidence level of 95%.

Discussion

It is evident that while **Model$_{\mathbf{inp}}$** demonstrates acceptable performance on the standard test set, the reverse is not true: the model trained on standard images fails to effectively generalize to inpainted ones. This can be explained by the fact that the model was trained on images with the KneeKG being a mobile part of the image with the tibial plate following the movement of the tibia and the femoral harness aligned with the knee orientation. The model likely learns to associate the orientation of those accessories with specific joint angle values; making them strong visual cues. Since the spatial configuration of these accessories is often correlated with knee flexion angles, the model is prone to overfitting if it focuses on locating the KneeKG and determine its orientation instead of focusing on intrinsic features like how the tibia and femur are oriented. Even with the introduction of the feature constraints we set, the installed KneeKG accessories remain an integral part of the lower right limb making their separation from the rest of the leg in feature selection challenging.
This assumption is supported by the feature maps plots of Figure 6.9. As expected, **Model$_{\mathbf{stand}}$**

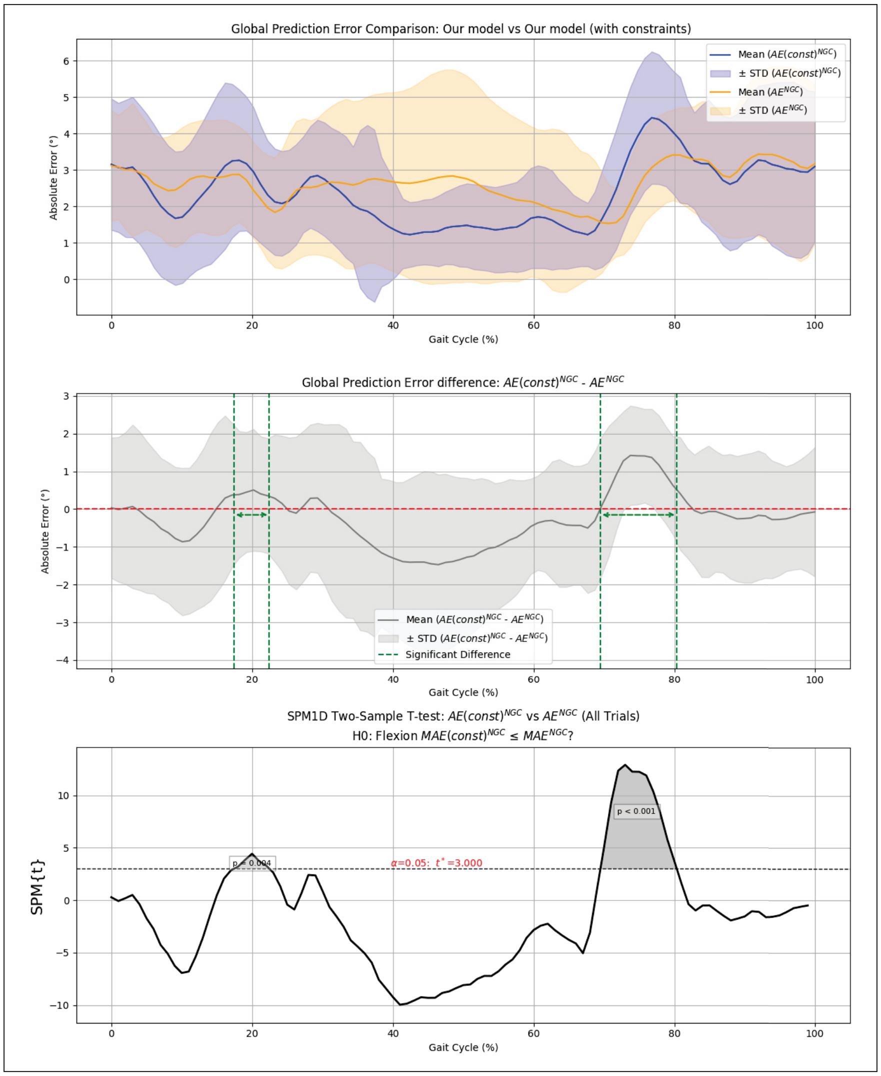


Figure 6.11 SMP1D-t: Does the constraint-based model perform better than the constraint-free model

is able to accurately locate the leg and its orientation with the KneeKG in the image. However, when the same image is inpainted, the model's ability to process the leg structure deteriorates. This explains the very high error in the angles predictions. In conclusion, the presence of KneeKG influences the learned representations and motivates the model to be dependent on these visual elements during inference.

Unlike **Model$_{\textbf{stand}}$**, **Model$_{\textbf{inp}}$** is trained on inpainted images which encourages the extraction of meaningful intrinsic features after the removal of the KneeKG accessories. The plots of

Figure 6.10 outline the ability of the model trained on inpainted images to locate the leg even with the presence of the KneeKG tool.

This section proved the importance of inpainting images in producing a less likely biased model that is prone to overfit.

According to the SPM results from Figure 6.11, we notice that H_0 is accepted on more than 85% of the gait cycle. In other terms, there is a probability of 95% that the model trained with constraints will perform better than the model without constraints on 85% of the gait cycle.

In this section, we proved a significant improvement on the test performance by adding the feature map constraints to the model.

Conclusion

In this chapter we provided an in-depth analysis of the global model and detailed performance. The patient and trial-wise results show the variability in the results due to the variety in anatomical properties of the participants.

Our model proved to have better results than other SOTA indirect angle estimation methods over the vast majority of the gait cycle.

The image inpainting proved its usefulness in making sure the trained model generalizes better and captures the leg intrinsic features better by removing the KneeKG relevance in the feature extraction.

The feature maps constraints help reduce the model overfitting during training and obtain a better validation/test performance. The repeated CV helps a less biased choice of the constraints coefficients over the four stages of the backbone.

CHAPTER 7

GENERAL DISCUSSION

In this study, we investigated whether it is feasible to estimate the knee joint kinematics with markerless techniques like HPE. The literature review provided a clear review of the current solutions and approaches used. Primarily the dominant idea is to rely on skeletal estimations from pretrained or retrained benchmark models to infer biomechanical parameters. Specifically, the knee flexion angle is one of the parameters of interest due to its importance in determining key gait cycle parameters for clinical assessment. The precision of the estimation and the clinical standard for the estimation bias are highly important to judge the methodology. In particular the gold standards for the clinically acceptable estimation errors is roughly 5° (McGinley *et al.*, 2009) with variations depending on the estimated parameter.

The motivation for this work arose from the observation that the flexion angle obtained through clinical measurement is closely aligned with the geometrical angle from the projected 2D joints coordinates in a lateral camera view setting. The pretrained pose estimation models like HRNet have shown difficulty in accurately estimating the 2D positions of the knee, ankle and femoral head joints for clinically acquired datasets. These datasets differ from standard benchmarks in terms of camera placement setting and joint annotation protocols. These factors result in relatively high coordinates errors. Consequently, calculating the flexion angle from these inaccurate coordinates predictions will yield an accumulated error and inconsistent angle estimation. These limitations and observations underscore the need for a more direct and robust approach to estimate 2D accessible parameters like the knee flexion angle from image data.

We collected a dataset from 30 different patients (25 for training and 5 for a holdout test set) doing walking trials. Those trials were conducted in a controlled laboratory environment with an equipped walking treadmill and the appropriate data acquisition tools, including two RGB cameras (frontal and lateral), an infrared camera, the KneeKG system and attached reflective markers. Through appropriate data processing and manipulation, we managed to create a synchronized image dataset with the appropriate annotations: the 2D and 3D joint coordinates, the three Euler angles and the transformation matrices. For this study, we exclusively relied on the images and the flexion angles and 2D positions annotations.

We proposed a deep neural network based on a pretrained convolutional backbone (ResNet) followed by a deep linear regression module to directly estimate the flexion angle without any intermediate joint coordinates estimations. Despite its architectural simplicity, the method performed effectively in a 2D monocular setting restricted to walking trials. It achieved largely acceptable clinical results outperforming other methods like HRNet (intermediate prediction based on joint coordinate estimations) and the 3D transformation estimation 6DRepnet. The results demonstrated a statistically significant improvement compared to the results of the mentioned SOTA approaches. The gait cycle -based SPM t-test proved that our model holds a lower error value at more than 90% of the NGC.

Our findings highlight the impact of image inpainting on model generalization. In fact, removing the KneeKG accessories from the images allows for a more task-relevant, determined and meaningful feature extraction unconditioned by the presence and orientation of the KneeKG. The feature maps of the model trained on the standard dataset failed to generalize when tested on inpainted images. The predicted angles were inaccurate and the model failed to extract the meaningful features from the image.

The introduction of feature maps constraints after certain intermediate outputs of the stages of the backbone proved a statistically significant improvement of the MAE of the at 85% of the NGC as validated by the SPM1D t-test results.

On a trial and participant level, we observed variability in the regions where the model performs best. However, we noticed a consistent pattern: the segment of the NGC between 35% and 65% consistently has the lowest estimation errors. This can be explained by the lower dynamic range of values of flexion angles in that segment of the gait cycle, where movement is more stable.

While this approach demonstrated strong performance according to the clinical standards with a relatively simple and light architecture, the hyperparameter tuning requires an extensive amount of training runs as finding the optimal configuration is resource-intensive. We had to limit the grid search space in terms of the feature maps constraints values to lower the high number of training runs.

The proposed pipeline proved its efficiency for the same action (walking activity) captured from fixed lateral camera in the same two controlled environments. Since it was not trained on more various scenarios (sequences captured from inclined cameras or other activities like squatting, running, a different background with different lighting, etc.), its generalization capacity to those conditions are expected to be limited. Although we attempted to perform some static actions like squatting but with very minimal movement, the limited range of motion made it impossible to produce enough variability in the images and angles data for an extensive feature extraction and proper learning.

Furthermore, although the repeated CV with a holdout test set proposes a fairly good number of validation scores to select an optimal architecture, this strategy is not without limitations. The performance of the model might still be influenced by the test set choice (choice of participants). In other terms, for a different random split we might get different results and different interpretations; possibly trials with lower or higher error values or different error distributions over the NGC.

Building on the findings of this work and following up on its limitations, future work could explore some promising directions. First, a more focused and more optimized hyperparameter selection method with a larger grid search space can be implemented.

Second, a nested CV can also be applied to assess the pipeline's performance on the whole dataset instead of a specific model instance on a random holdout test set. Although it comes with a much higher cost, it ensures a non-biased evaluation to a certain extent.

Moreover, to test the model's capacity to predict the angles directly from image-based sources in different settings, the dataset can be extended to include a more diverse set of camera positions introducing depth variations and changes in the sagittal plane alignment. These factors are significantly influential on the visual appearance and can affect the accuracy of the estimate. In addition, it is possible to include other different actions like running, active squatting (going from a standing to a full-depth squat position), etc allowing the model to learn a larger range of motion patterns. Training and testing the model on these actions both separately and in combination will yield more concise insights regarding the generalization potential of the pipeline.

CONCLUSION AND RECOMMENDATIONS

This thesis explored a markerless deep learning approach for the estimation of the knee flexion angles during walking. From a single monocular RGB image of a person during walking activity captured from lateral view, the model is able to predict the knee flexion angle with an average of 2.4° error value.
Motivated by the limitations of the generic pose estimation models in clinical settings, we proposed a direct regression pipeline based on a simple yet effective convolutional architecture. The highlight of this work is the estimation of the flexion angle without relying on intermediate joint coordinate predictions, avoiding any error accumulations.

The dataset composed of multiple participants was constructed resulting in synchronized video sequences with the appropriate recorded annotations. For the purpose of this study, the model was trained exclusively on RGB images of walking trials with the flexion angle and 2D joint coordinates estimations.

The pretrained ResNet-based backbone followed by a linear regression head demonstrated clinically acceptable performance. It outperformed SOTA methods such as HRNet (keypoint's estimation) and 6DRepNet (orientation estimation) in terms of the flexion angle estimation. SPM analysis confirmed the significance of this improvement in the vast majority of the cycle.

Several enhancements contributed to the performance improvement (AE reduction), notably the introduction of feature map constraints at intermediate stags of the backbone to guide the learning process and condition it. The role of image inpainting was also emphasized by a quantitative (error analysis) and qualitative (feature maps visualization) assessment. It was concluded that the image inpainting by removing the KneeKG is crucial in effectively training a model to extract meaningful features.

APPENDIX I

GROUND-TRUTH, PREDICTED ANGLES AND ERRORS DISTRIBUTION OF THE MODELS TRAINED ON STANDARD AND ON INPAINTED DATASETS

1. Model$_{stand}$ histograms

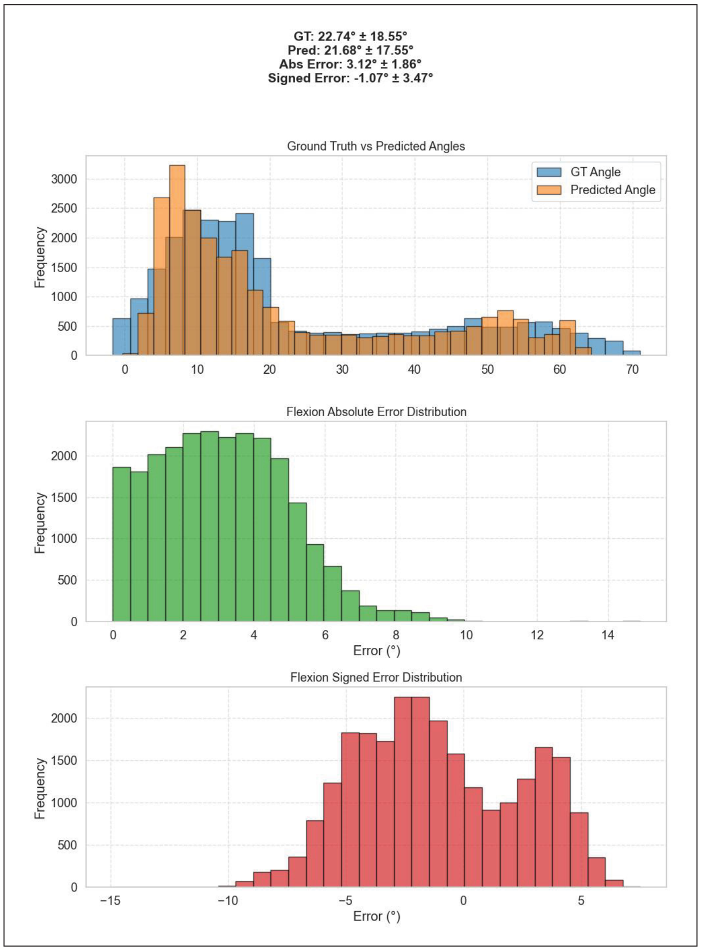
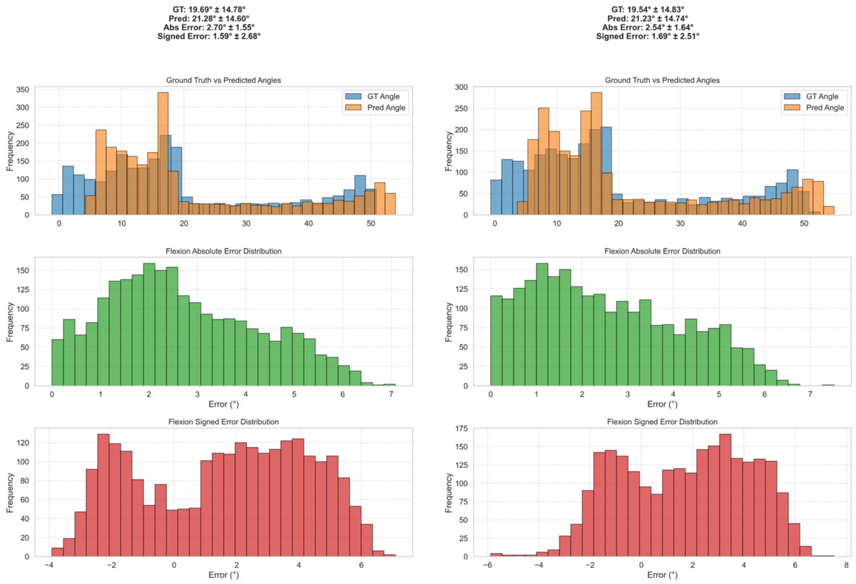


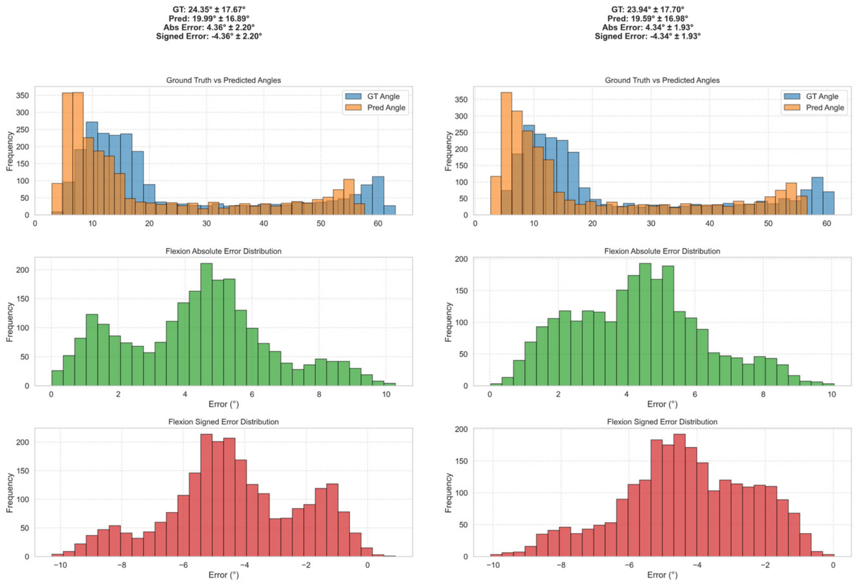
Figure-A I-1 GT, predictions and error distribution of all test trials



a) Trial 001

b) Trial 002

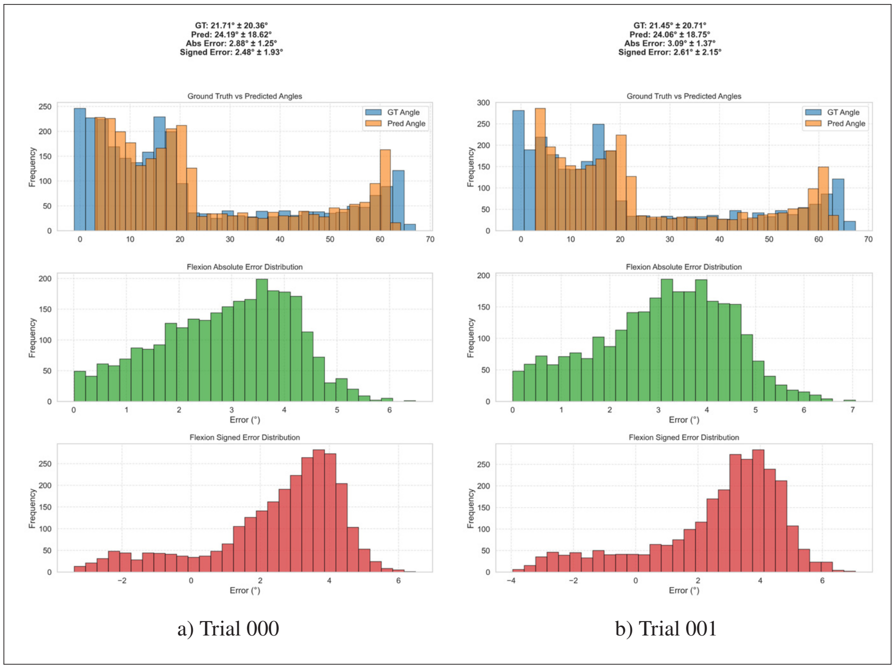
Figure-A I-2 GT, predictions and error distribution of Participant P05



a) Trial 001

b) Trial 002

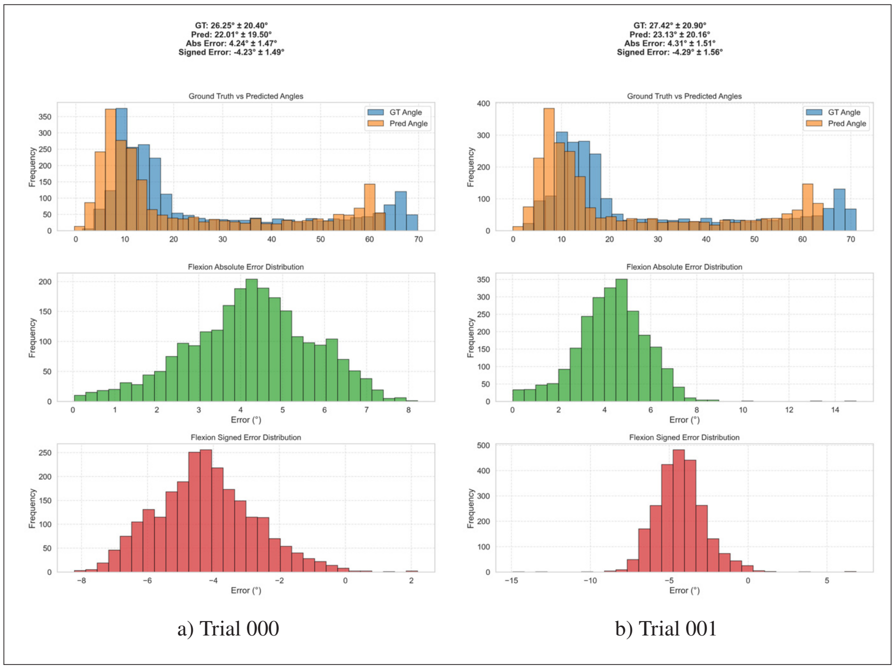
Figure-A I-3 GT, predictions and error distribution of Participant P06



a) Trial 000

b) Trial 001

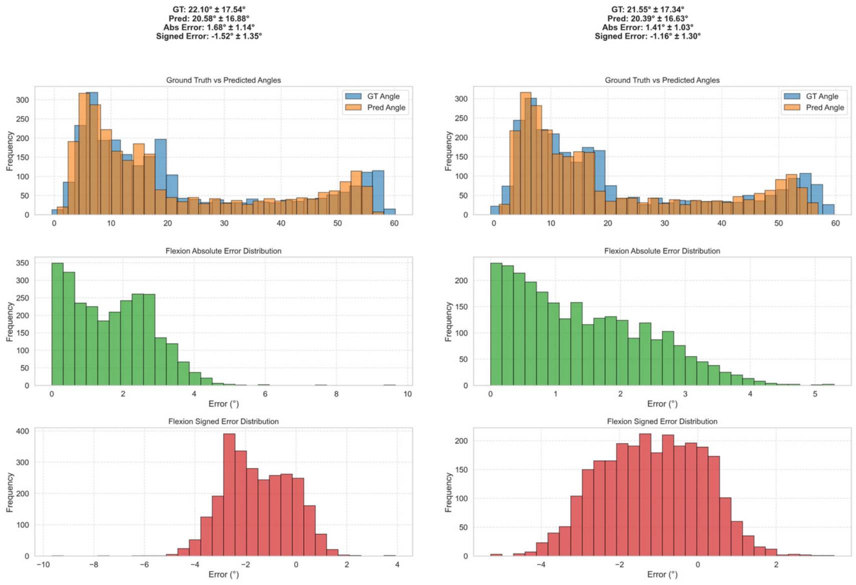
Figure-A I-4 GT, predictions and error distribution of Participant P15E



a) Trial 000

b) Trial 001

Figure-A I-5 GT, predictions and error distribution of Participant P20



a) Trial 001

b) Trial 002

Figure-A I-6 GT, predictions and error distribution of Participant P24

2. Model$_{inp}$ histograms

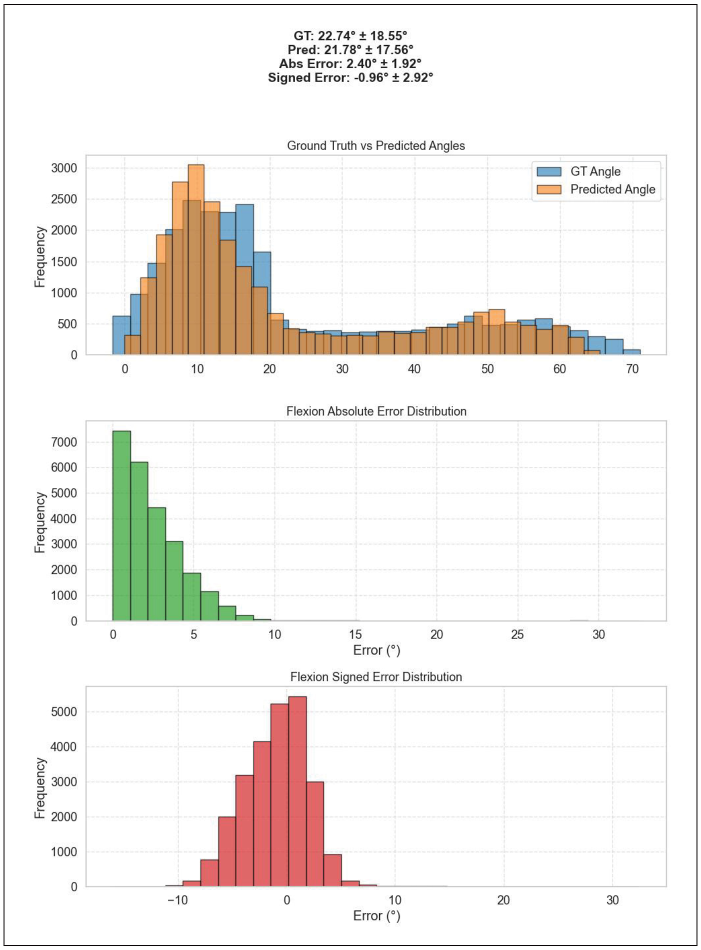
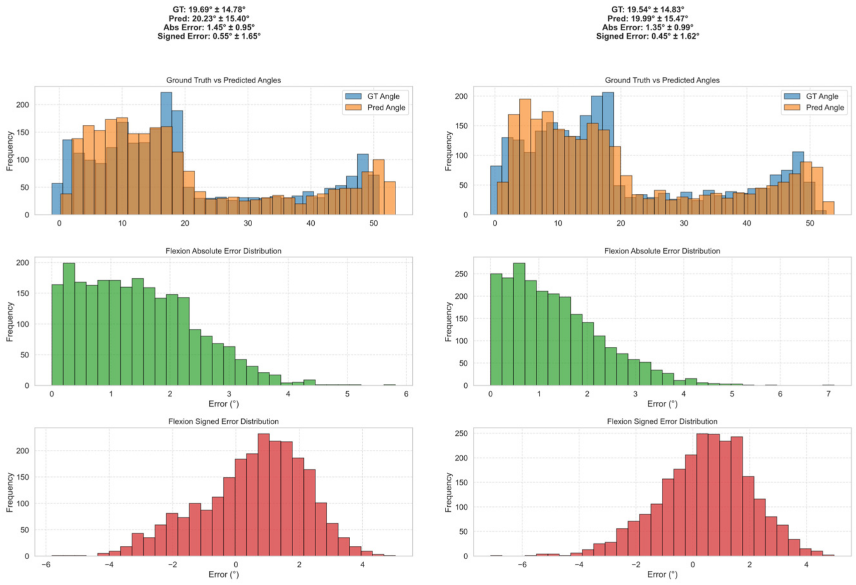


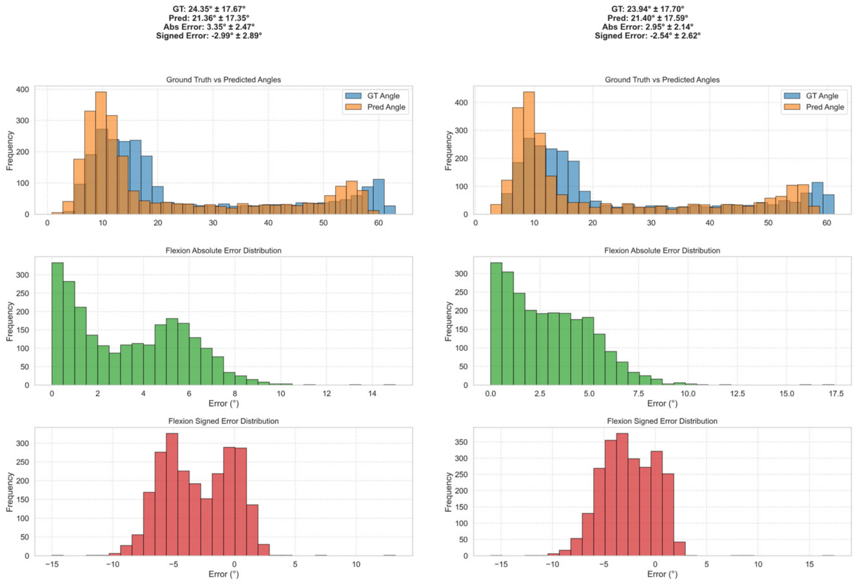
Figure-A I-7 GT, predictions and error distribution of all test trials



a) Trial 001

b) Trial 002

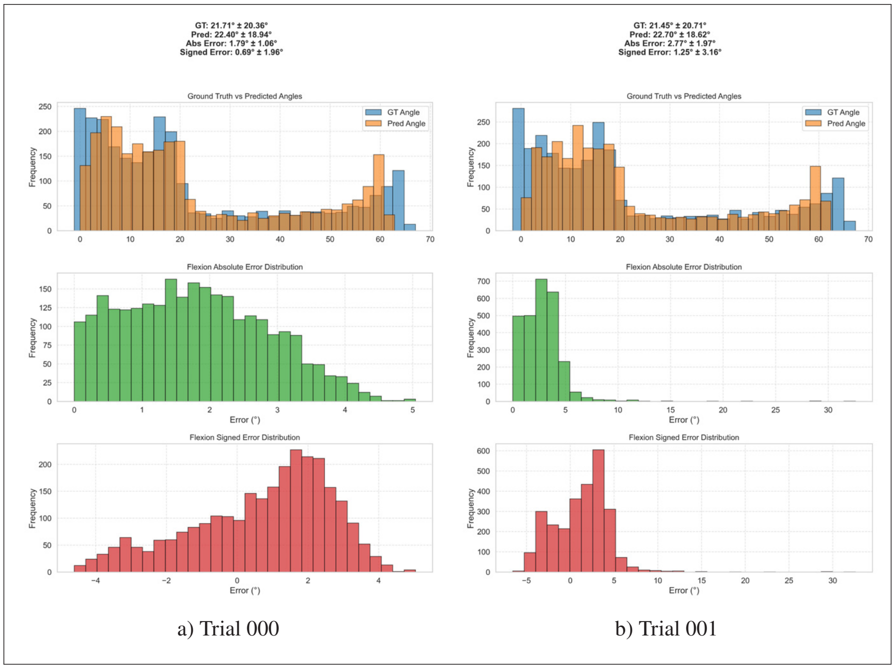
Figure-A I-8 GT, predictions and error distribution of Participant P05



a) Trial 001

b) Trial 002

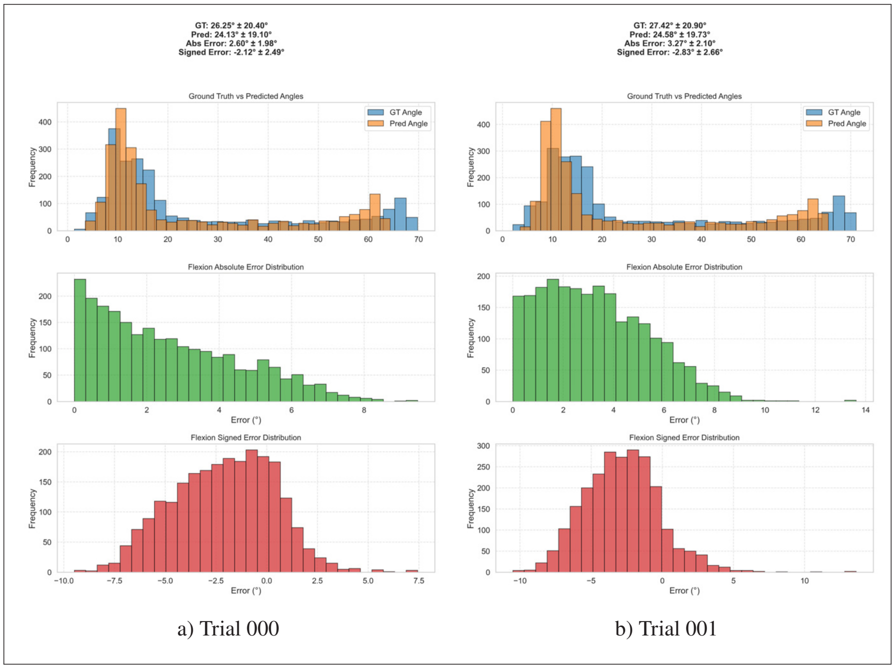
Figure-A I-9 GT, predictions and error distribution of Participant P06



a) Trial 000

b) Trial 001

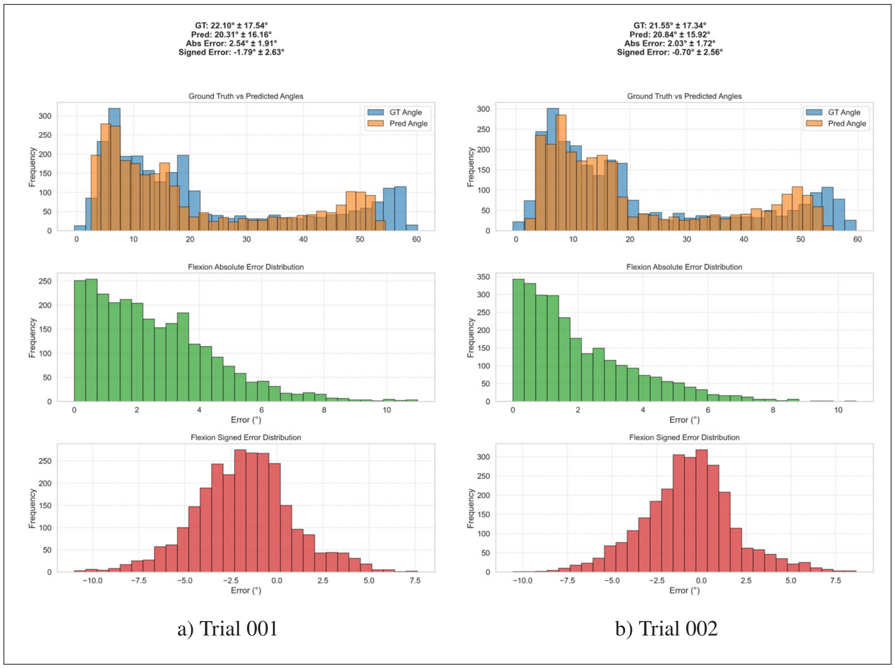
Figure-A I-10 GT, predictions and error distribution of Participant P15E



a) Trial 000

b) Trial 001

Figure-A I-11 GT, predictions and error distribution of Participant P20



a) Trial 001

b) Trial 002

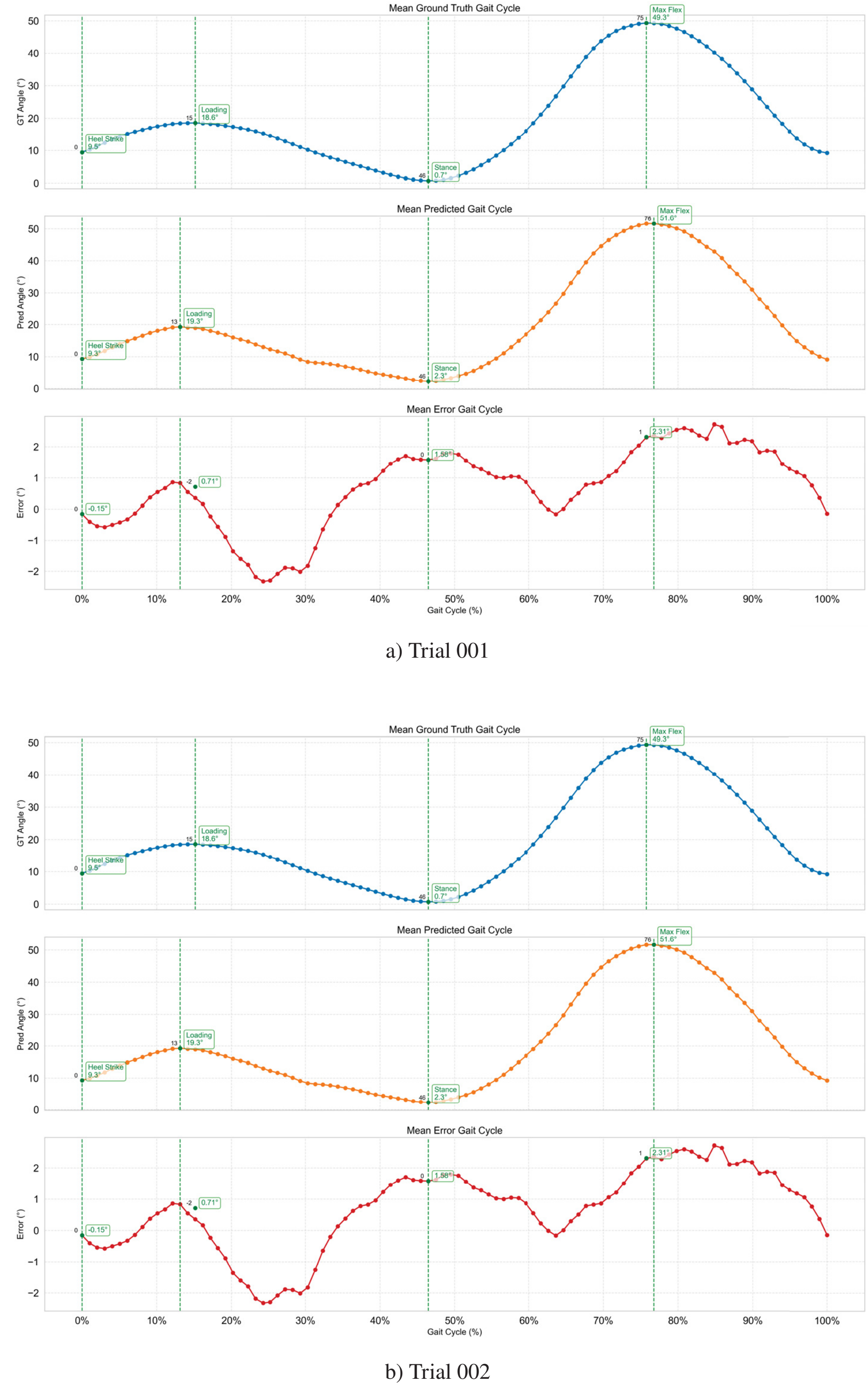
Figure-A I-12 GT, predictions and error distribution of Participant P24

APPENDIX II

MEAN NORMALIZED GAIT CYCLES

1. Trial mean gait cycles

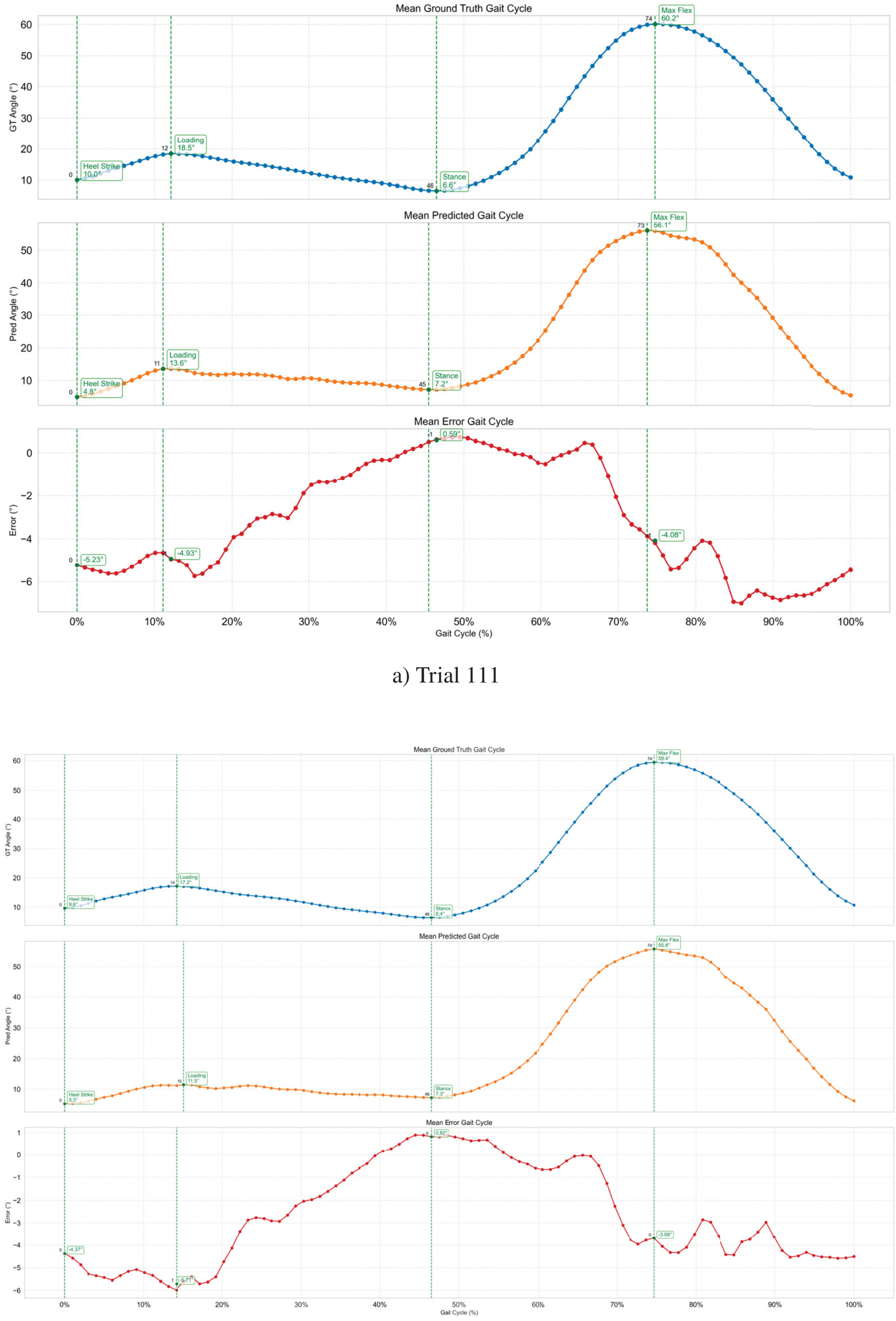
2. Participant mean gait cycles



a) Trial 001

b) Trial 002

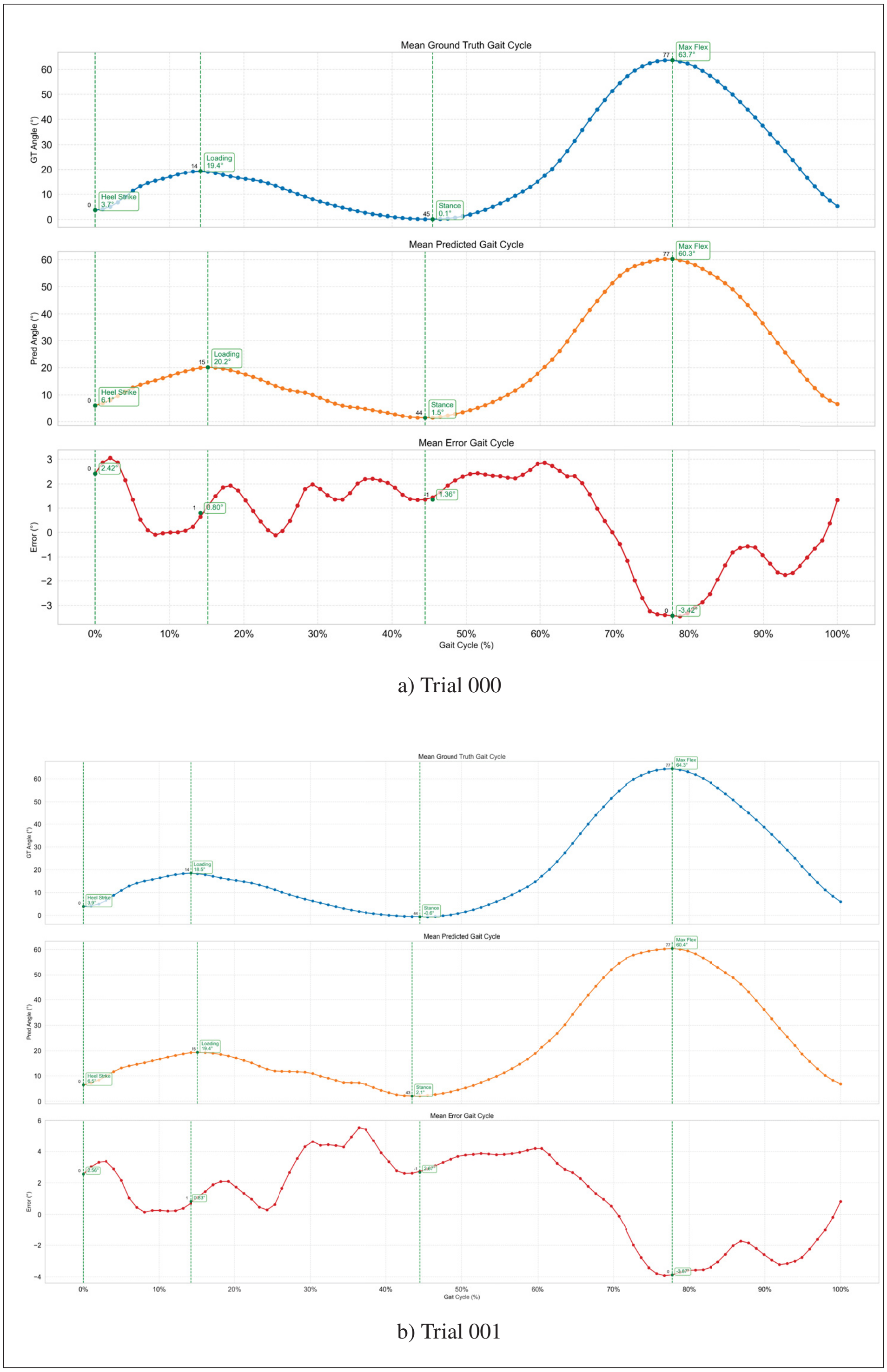
Figure-A II-1 Mean NGCs of Participant P05



a) Trial 111

b) Trial 112

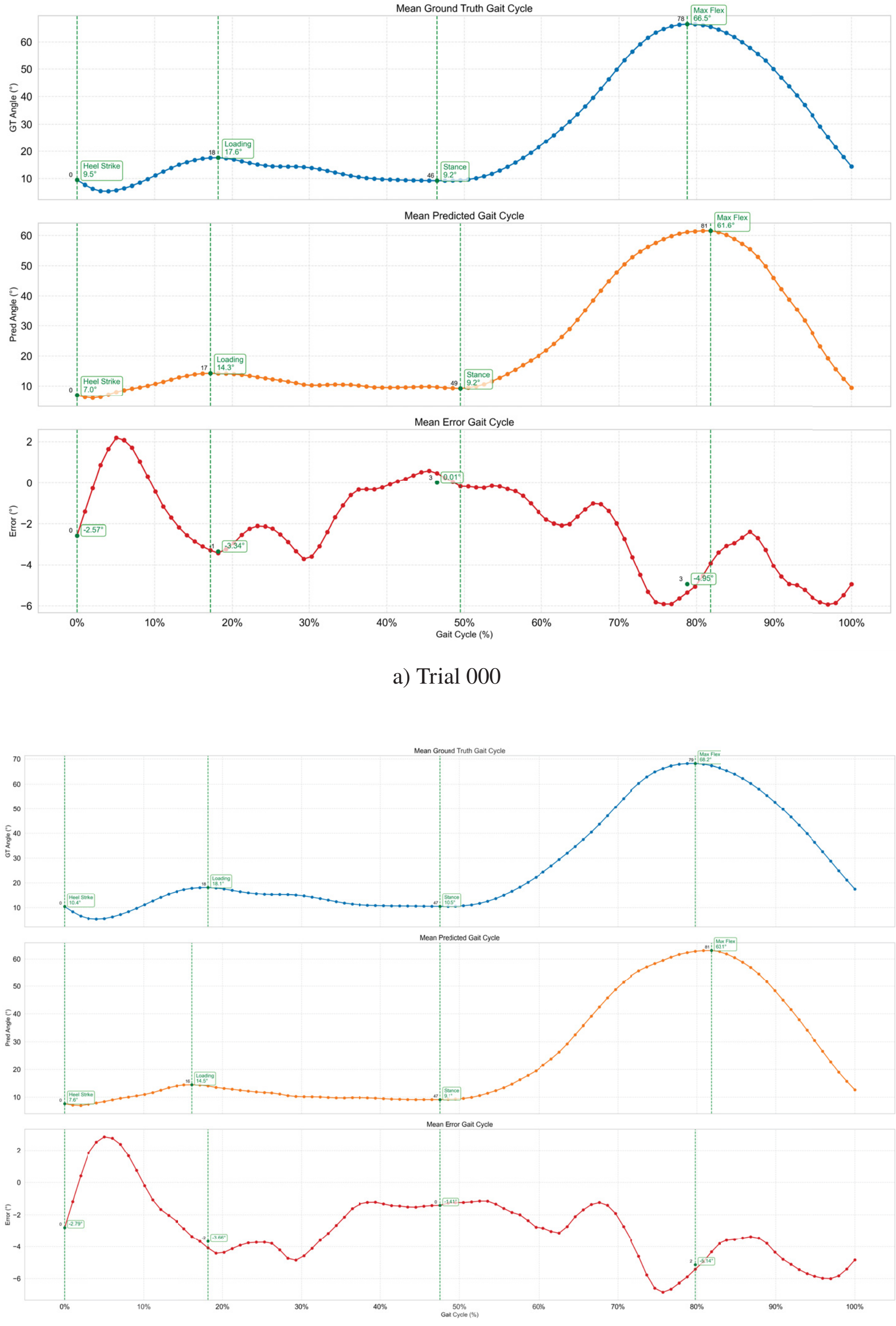
Figure-A II-2 Mean NGCs of Participant P06



a) Trial 000

b) Trial 001

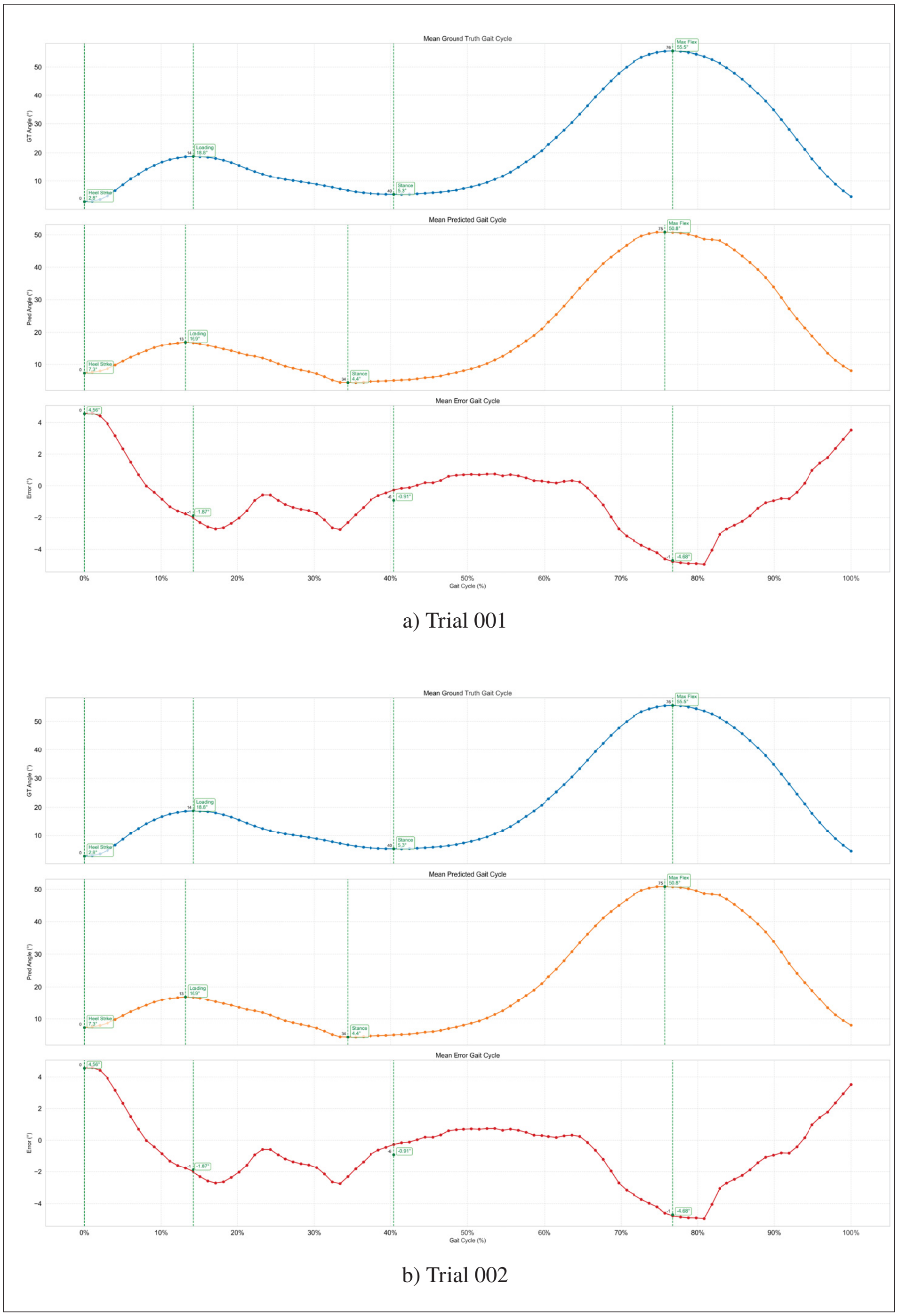
Figure-A II-3 Mean NGCs of Participant P15E



a) Trial 000

b) Trial 001

Figure-A II-4 Mean NGCs of Participant P20



a) Trial 001

b) Trial 002

Figure-A II-5   Mean NGCs of Participant P24

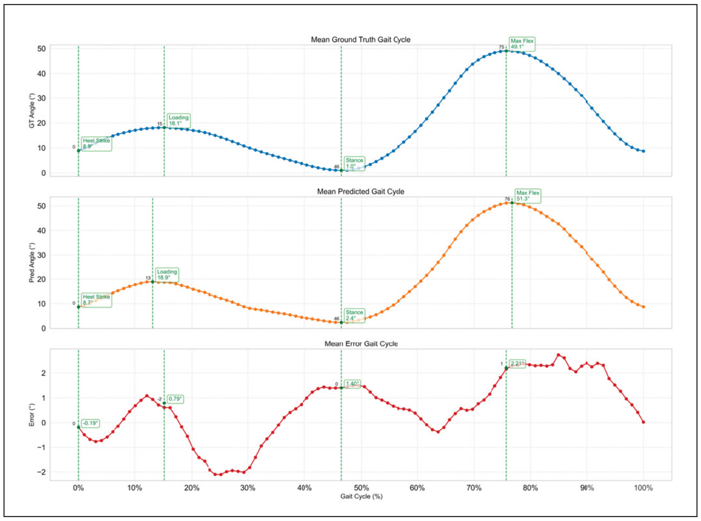


Figure-A II-6 Mean NGCs of Participant P05 (both trials)

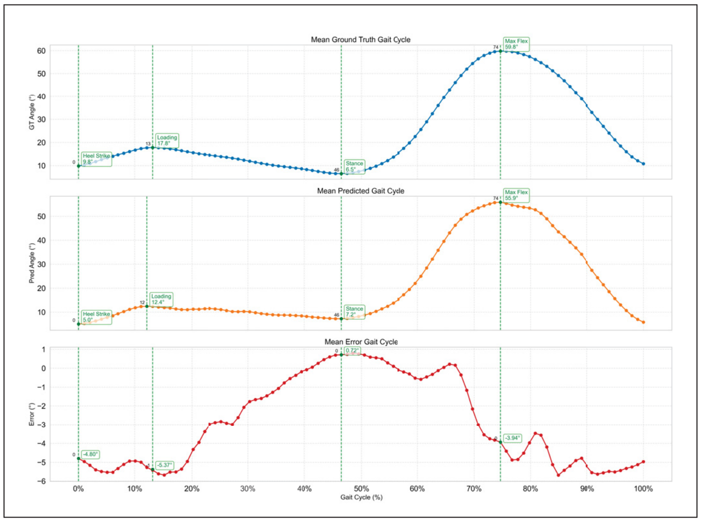


Figure-A II-7 Mean NGCs of Participant P06 (both trials)

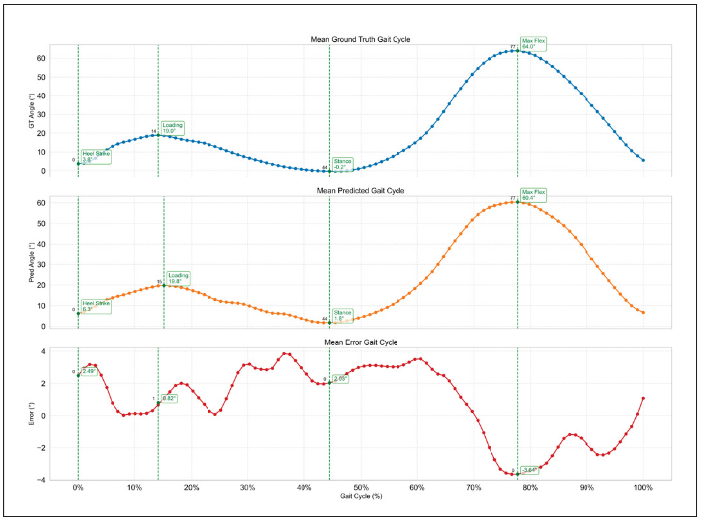


Figure-A II-8 Mean NGCs of Participant P15E (both trials)

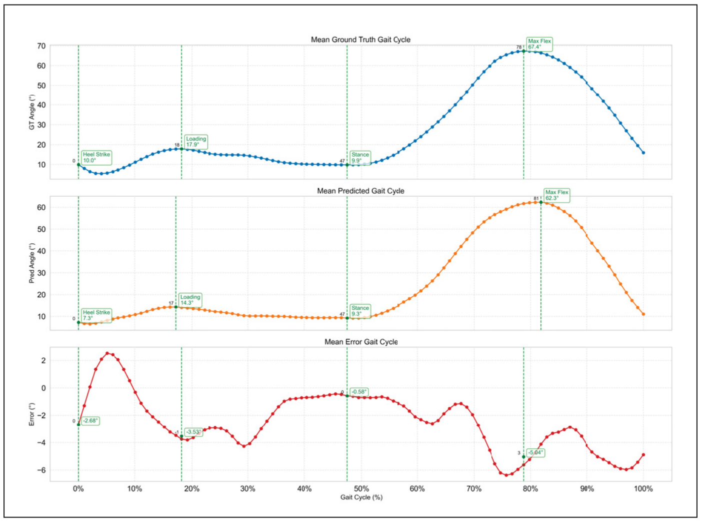


Figure-A II-9 Mean NGCs of Participant P20 (both trials)

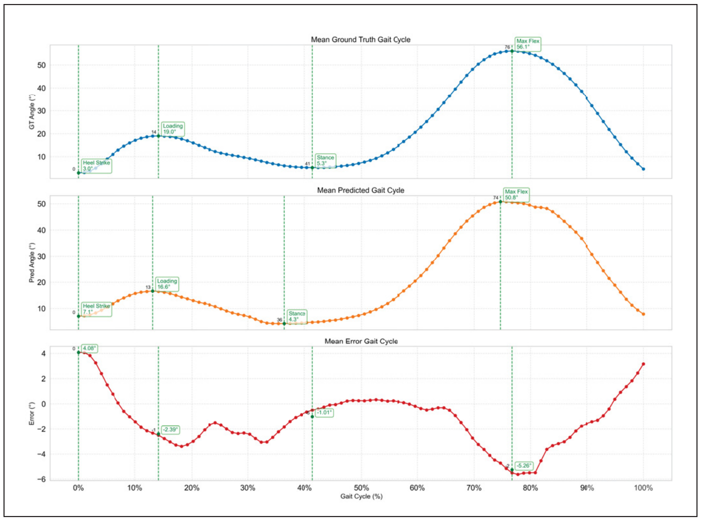
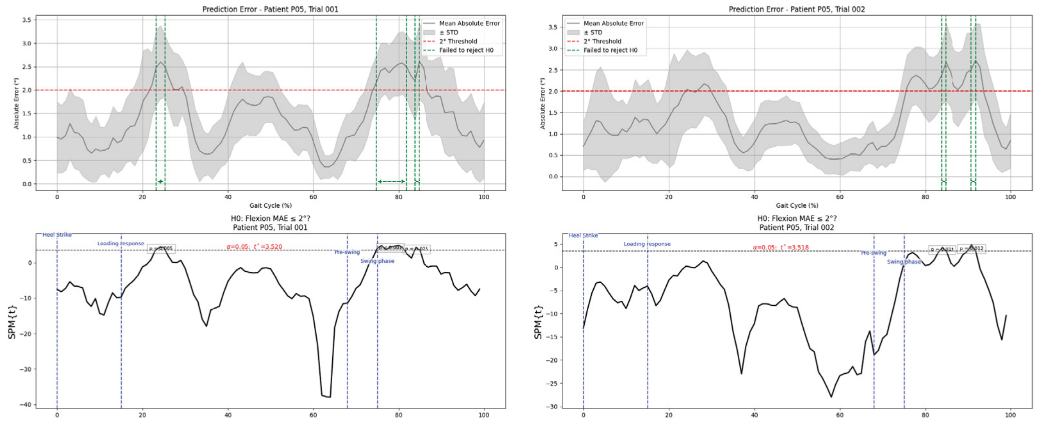


Figure-A II-10 Mean NGCs of Participant P24 (both trials)

APPENDIX III

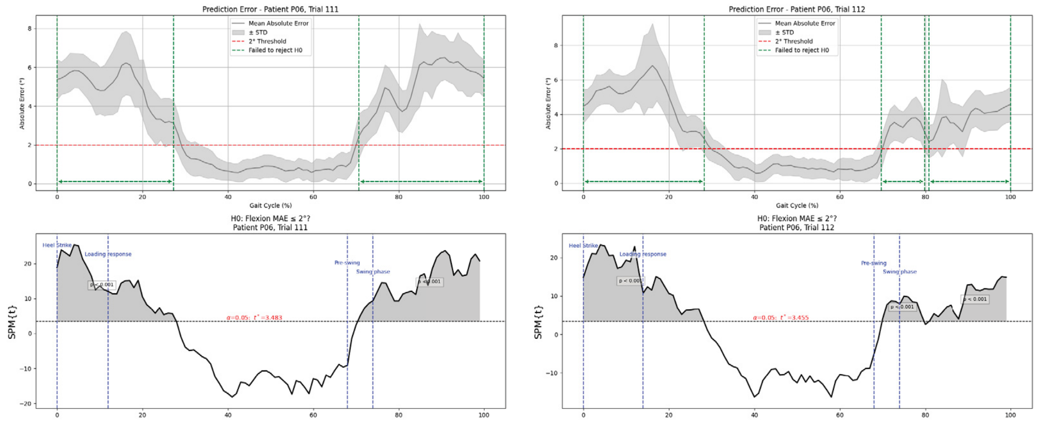
STATISTICAL PARAMETRIC MAPPING TESTS

1. Trial tests



a) Trial 001 b) Trial 002

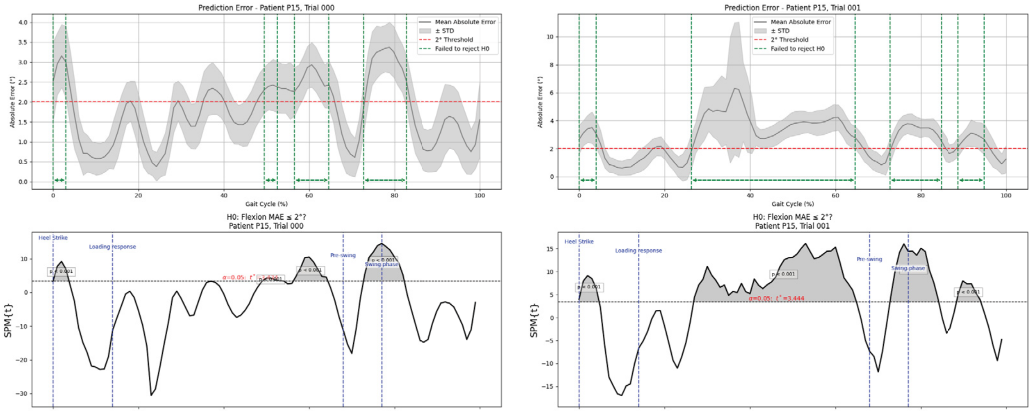
Figure-A III-1 SPM1D Test: difference between AE gait cycles (P05)



a) Trial 111 b) Trial 112

Figure-A III-2 SPM1D Test: difference between AE gait cycles (P06)

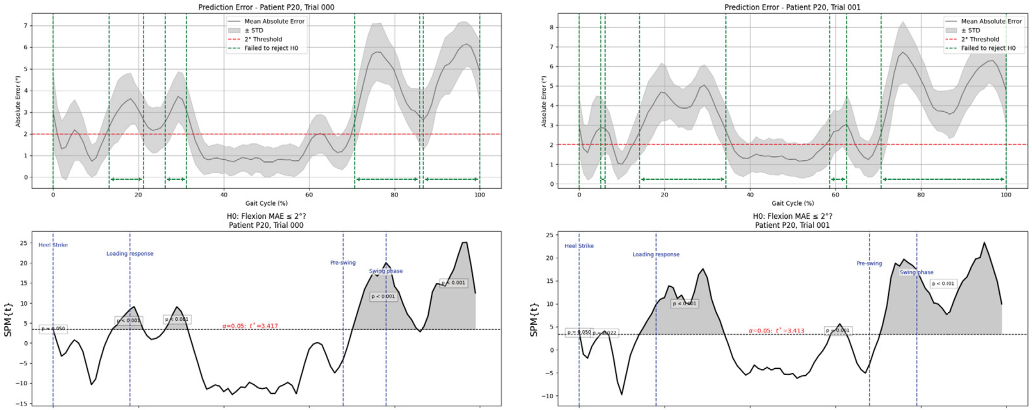
2. Participant tests



a) Trial 000

b) Trial 001

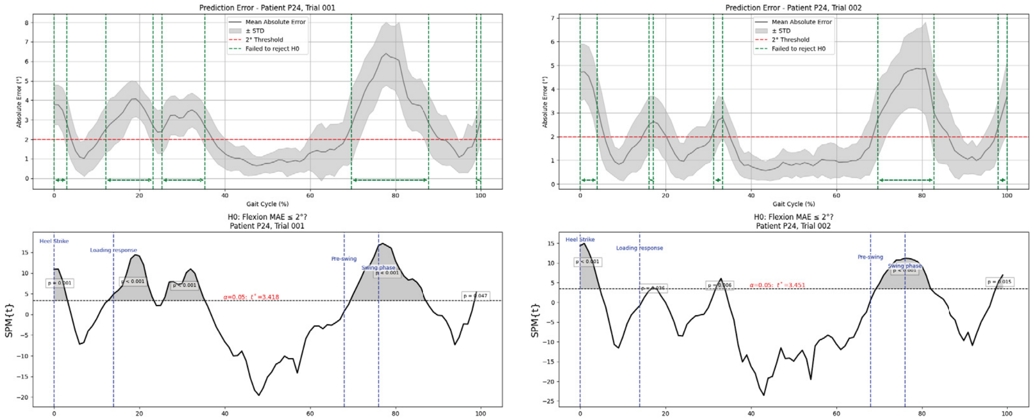
Figure-A III-3   SPM1D Test: the difference between AE gait cycles (P15E)



a) Trial 000

b) Trial 001

Figure-A III-4   SPM1D Test: the difference between AE gait cycles (P20)



a) Trial 000

b) Trial 001

Figure-A III-5 SPM1D Test: the difference between AE gait cycles (P24)

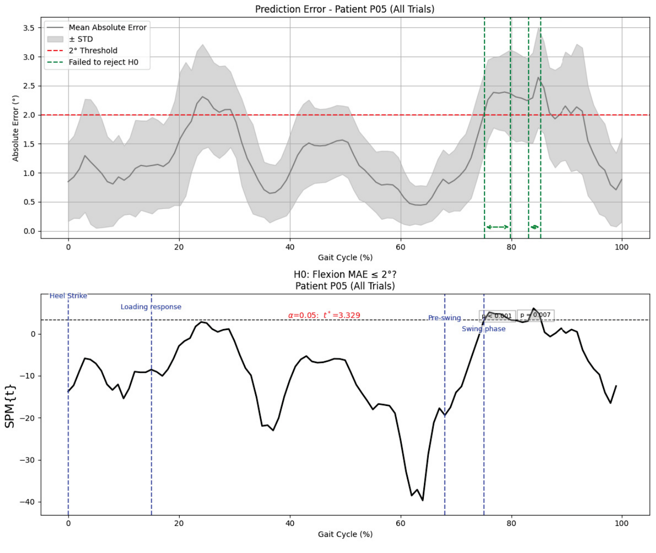


Figure-A III-6 SPM1D Test: the difference between AE gait cycles (P05 - all trials)

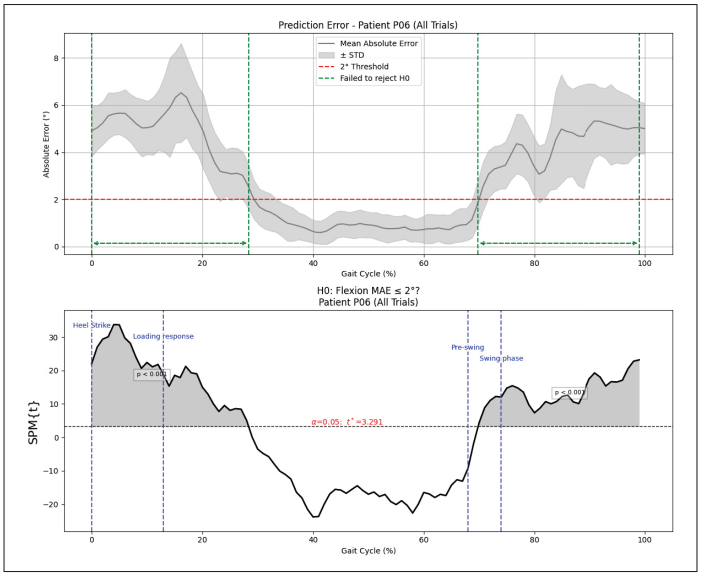


Figure-A III-7 SPM1D Test: the difference between AE gait cycles (P06 - all trials)

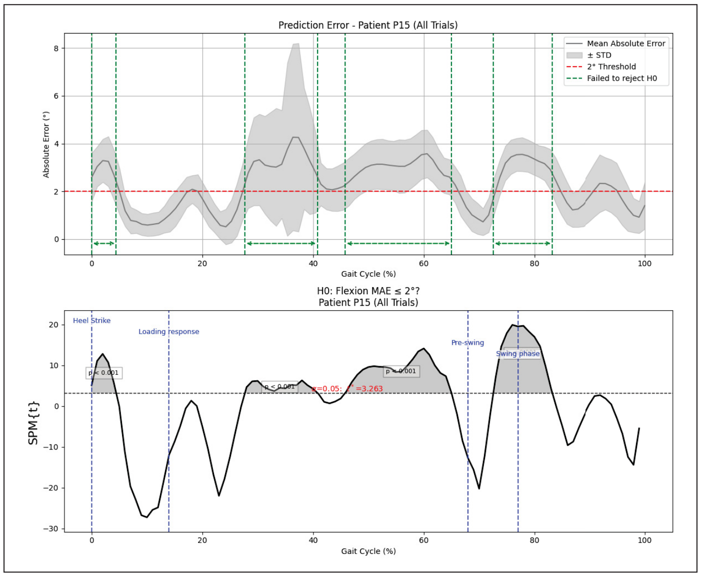


Figure-A III-8 SPM1D Test: the difference between AE gait cycles (P15E - all trials)

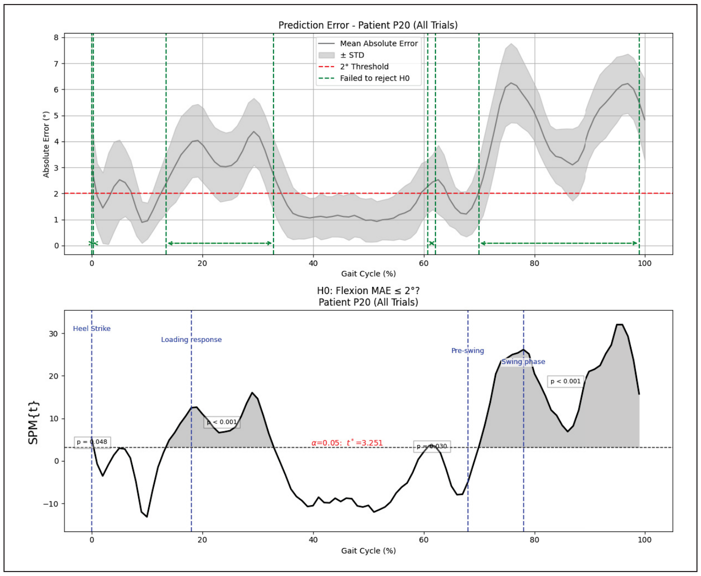


Figure-A III-9 SPM1D Test: the difference between AE gait cycles (P20 - all trials)

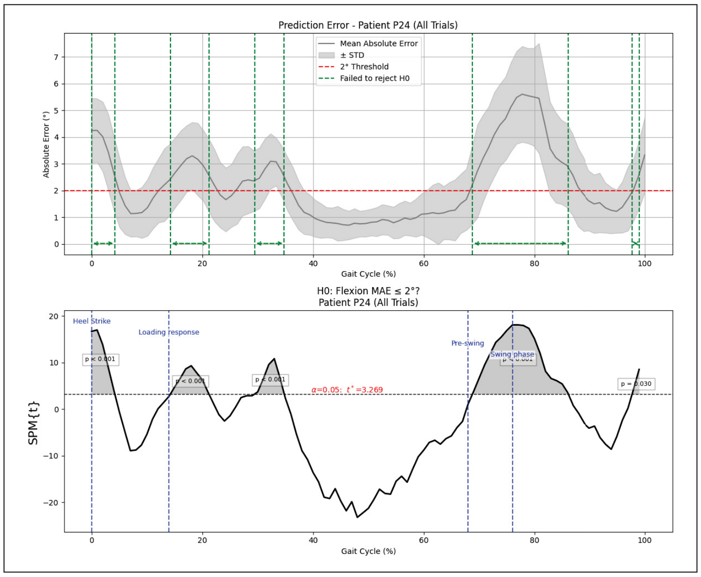


Figure-A III-10 SPM1D Test: the difference between AE gait cycles (P24 - all trials)

APPENDIX IV

CROSS-VALIDATION RESULTS

1. Cross-validation results for standard dataset

Table-A IV-1 Validation MAE (MAE) for RepVGG backbone network with different hyperparameters (λ) configurations across multiple shuffle splits and CV folds (standard dataset)

Backbone	$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	**Validation MAE**												**AVG**
		Shuffle Split 1				**Shuffle Split 2**				**Shuffle Split 3**				
		CV 1	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	
RepVGG	(0,0,0,0)	3.25	2.05	2.73	2.50	2.81	2.67	2.63	2.17	2.06	2.21	2.42	3.07	2.5475
	(1000,0,0,0)	3.74	1.98	2.78	2.25	2.68	2.58	2.64	3.01	2.52	2.12	2.51	3.13	2.661667
	(0,1000,0,0)	2.93	1.93	2.31	2.60	2.74	2.56	2.15	2.69	2.26	2.21	2.03	2.88	2.440833
	(0,0,1000,0)	3.16	2.04	2.49	2.46	2.78	2.48	2.40	2.05	2.27	2.42	1.98	2.69	2.4350
	(0,0,0,1000)	2.79	2.21	2.29	2.43	2.49	2.43	2.24	2.24	2.25	2.31	2.08	2.63	2.365833
	(1000,1000,0,0)	3.36	2.15	2.53	3.06	2.60	2.62	2.41	3.54	2.97	2.29	2.49	3.08	2.758333
	(1000,0,1000,0)	3.42	2.19	2.57	2.39	2.62	2.42	2.41	2.59	2.28	2.48	2.33	2.84	2.5450
	(1000,0,0,1000)	3.11	2.19	2.44	2.41	2.60	2.42	2.19	2.58	2.38	2.26	2.41	2.88	2.489167
	(0,1000,1000,0)	2.92	2.01	2.29	2.55	2.43	2.56	2.26	1.98	2.30	2.35	2.13	2.69	2.3725
	(0,1000,0,1000)	2.97	2.16	2.08	3.02	2.64	2.54	2.07	2.54	2.41	2.21	2.28	2.78	2.4750
	(0,0,1000,1000)	2.76	1.94	2.41	2.49	2.48	2.55	2.35	1.84	2.41	2.21	2.19	2.66	2.3575
	(1000,1000,1000,0)	2.97	2.31	2.05	2.67	2.42	2.63	2.33	2.93	2.62	2.30	2.05	3.12	2.533333
	(1000,1000,0,1000)	3.29	2.25	2.52	3.24	2.99	2.49	2.20	3.27	2.75	2.36	2.94	2.93	2.769167
	(1000,0,1000,1000)	3.36	2.11	2.40	2.42	2.31	2.30	2.12	2.68	2.40	2.37	2.34	2.87	2.473333
	(0,1000,1000,1000)	2.68	2.01	2.15	2.57	2.53	2.63	2.41	2.09	2.40	2.08	2.30	2.83	2.3900
	(1000,1000,1000,1000)	2.98	2.24	2.68	3.05	2.80	2.47	2.02	3.11	2.61	2.06	3.06	2.81	2.6575

Table-A IV-2 Validation MAE (MAE) for ResNet backbone network with different hyperparameter (λ) configurations across multiple shuffle splits and CV folds (standard dataset)

Backbone	$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	**Validation MAE**												**AVG**
		Shuffle Split 1				**Shuffle Split 2**				**Shuffle Split 3**				
		CV 1	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	
Resnet	(0,0,0,0)	2.96	2.27	2.16	2.19	2.79	2.46	2.38	2.30	2.42	2.36	2.50	2.38	2.430833
	(1000,0,0,0)	2.75	2.21	1.91	2.37	3.22	2.33	2.23	1.97	2.68	2.08	2.14	2.63	2.376667
	(0,1000,0,0)	2.60	2.14	1.96	2.69	3.03	2.23	2.07	2.05	2.24	2.00	2.12	2.58	2.309167
	(0,0,1000,0)	2.30	2.11	1.73	2.37	2.45	2.15	2.17	1.94	2.35	2.23	2.48	2.33	2.2175
	(0,0,0,1000)	2.54	2.13	1.75	2.75	2.73	2.27	2.26	2.19	2.42	2.15	2.64	2.14	2.330833
	(1000,1000,0,0)	2.67	1.82	1.99	2.54	2.37	2.23	2.10	1.77	3.13	1.82	1.99	2.73	2.263333
	(1000,0,1000,0)	2.19	2.02	1.73	2.67	2.50	2.26	2.17	1.78	2.83	2.23	2.04	2.41	2.235833
	(1000,0,0,1000)	2.55	2.15	1.80	2.53	2.25	2.16	2.23	1.75	2.52	2.14	2.20	2.31	**2.215833**
	(0,1000,1000,0)	2.26	2.07	1.89	2.95	2.88	2.14	2.02	1.61	2.45	1.93	2.18	2.37	2.229167
	(0,1000,0,1000)	2.64	2.10	1.90	3.00	2.95	2.27	2.25	1.71	2.44	2.03	2.85	2.57	2.3925
	(0,0,1000,1000)	2.44	2.05	1.58	2.82	2.82	2.30	2.02	1.72	2.39	2.20	2.54	2.01	2.240833
	(1000,1000,1000,0)	2.44	1.91	1.71	2.62	2.33	2.23	2.07	1.83	2.80	1.93	2.18	2.60	2.220833
	(1000,1000,0,1000)	3.02	2.00	1.65	2.95	2.27	2.40	2.43	2.04	2.57	1.89	2.39	2.58	2.349167
	(1000,0,1000,1000)	2.35	2.17	1.67	2.95	2.54	2.37	2.02	1.80	2.31	2.21	2.04	2.44	2.239167
	(0,1000,1000,1000)	2.49	2.08	1.58	2.62	3.32	2.15	2.08	1.60	2.39	1.93	2.10	2.26	2.216667
	(1000,1000,1000,1000)	2.55	2.07	1.73	2.80	2.39	2.31	2.18	1.68	2.62	1.92	2.01	2.47	2.2275

2. Cross-validation results for inpainted dataset

Table-A IV-3 Validation MAE (MAE) for RepVGG backbone network with different hyperparameter (λ) configurations across multiple shuffle splits and CV folds (Inpainted dataset)

Backbone	$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	**Validation MAE**												**AVG**
		Shuffle Split 1				**Shuffle Split 2**				**Shuffle Split 3**				
		CV 1	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	
RepVGG	(0,0,0,0)	3.80	2.26	3.83	3.37	3.10	3.21	2.44	3.87	3.70	2.62	2.45	4.00	3.22
	(1000,0,0,0)	4.37	2.44	3.73	3.46	3.34	3.23	2.96	4.15	4.23	2.75	2.72	4.21	3.47
	(0,1000,0,0)	3.66	2.30	3.13	3.55	2.89	3.31	2.35	3.56	3.89	2.61	2.18	3.97	3.12
	(0,0,1000,0)	3.76	2.53	2.84	2.94	3.11	3.02	2.41	3.78	3.41	2.88	2.60	3.65	3.08
	(0,0,0,1000)	3.43	2.39	3.18	3.42	2.97	2.92	2.41	3.51	3.96	2.69	2.06	3.61	3.05
	(1000,1000,0,0)	4.35	2.29	3.30	3.86	3.15	3.27	2.82	4.93	3.87	2.72	2.65	3.85	3.42
	(1000,0,1000,0)	4.13	2.42	3.08	3.42	3.08	3.00	2.42	4.78	3.54	2.65	2.73	3.81	3.26
	(1000,0,0,1000)	3.72	2.52	3.13	3.59	3.10	3.01	2.24	3.88	4.03	2.72	2.27	3.88	3.17
	(0,1000,1000,0)	3.35	2.60	3.30	3.49	3.25	3.32	2.39	4.09	3.58	2.51	2.58	3.79	3.19
	(0,1000,0,1000)	3.33	2.51	2.75	3.84	3.16	3.18	2.25	3.90	3.66	2.43	2.64	3.66	3.11
	(0,0,1000,1000)	3.67	2.46	3.07	3.21	3.05	3.01	2.48	3.54	3.30	2.81	2.56	3.82	3.08
	(1000,1000,1000,0)	3.67	2.53	2.84	3.99	3.18	3.27	2.27	3.97	3.74	2.68	2.38	3.66	3.18
	(1000,1000,0,1000)	4.06	2.53	3.18	4.33	2.95	3.16	2.38	4.64	4.14	2.45	2.62	3.57	3.33
	(1000,0,1000,1000)	3.88	2.44	3.17	3.48	2.99	3.11	2.36	3.99	3.64	2.78	2.64	3.71	3.18
	(0,1000,1000,1000)	3.23	2.46	3.06	3.83	3.12	3.22	2.37	3.35	3.91	2.54	2.49	3.78	3.11
	(1000,1000,1000,1000)	3.87	2.61	3.16	4.06	2.97	3.22	2.44	3.76	3.57	2.55	2.96	3.52	3.22

Table-A IV-4 Validation MAE (MAE) for ResNet backbone network with different hyperparameter (λ) configurations across multiple shuffle splits and CV folds (Inpainted dataset)

Backbone	$(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	**Validation MAE**												**AVG**
		Shuffle Split 1				**Shuffle Split 2**				**Shuffle Split 3**				
		CV 1	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	**CV 1**	**CV 2**	**CV 3**	**CV 4**	
ResNet	(0,0,0,0)	3.57	2.44	2.94	3.50	3.33	3.35	2.50	3.44	2.95	2.60	2.70	3.70	3.09
	(1000,0,0,0)	3.53	2.52	2.72	3.63	3.10	2.98	2.49	3.70	3.30	2.25	2.27	3.75	3.02
	(0,1000,0,0)	3.88	2.64	2.28	4.11	3.23	3.36	2.26	3.18	3.29	2.49	3.05	3.96	3.14
	(0,0,1000,0)	3.05	2.55	2.62	4.36	3.21	3.09	2.17	3.35	3.08	2.54	2.48	4.16	3.06
	(0,0,0,1000)	3.33	2.39	2.77	3.66	3.24	2.98	2.19	4.34	3.32	2.61	2.64	3.72	3.10
	(1000,1000,0,0)	3.53	2.66	2.30	3.55	3.03	3.29	2.28	3.42	3.50	2.36	2.65	3.62	3.02
	(1000,0,1000,0)	3.12	2.61	2.62	3.62	3.14	3.10	2.17	3.16	3.56	2.50	2.46	3.46	2.96
	(1000,0,0,1000)	3.18	2.42	2.50	3.97	2.95	2.91	2.32	3.86	4.40	2.33	2.17	3.53	3.05
	(0,1000,1000,0)	3.30	2.93	2.53	4.23	3.37	3.13	2.29	3.13	3.42	2.41	3.20	3.80	3.15
	(0,1000,0,1000)	3.32	2.51	2.16	3.62	3.16	3.15	2.20	3.14	3.33	2.13	2.56	3.55	**2.90**
	(0,0,1000,1000)	3.13	2.44	2.86	4.21	3.26	2.98	2.36	3.42	3.12	2.64	2.68	4.14	3.10
	(1000,1000,1000,0)	3.37	2.65	2.16	3.82	3.45	3.22	2.11	3.45	3.51	2.60	2.81	3.72	3.07
	(1000,1000,0,1000)	3.40	2.54	2.39	4.17	2.85	2.99	2.32	3.77	4.01	2.35	2.49	3.78	3.09
	(1000,0,1000,1000)	3.21	2.74	2.79	3.80	3.10	3.00	2.07	3.32	3.49	2.38	2.48	3.97	3.03
	(0,1000,1000,1000)	3.26	2.80	2.59	4.64	3.64	3.59	2.35	3.19	3.63	2.68	3.16	4.22	3.31
	(1000,1000,1000,1000)	3.58	2.79	2.50	3.86	3.37	3.15	2.36	3.81	3.67	2.43	2.76	3.70	3.17

APPENDIX V

DETAILED ACQUISITION PROCEDURE

This appendix details the step-by-step procedure followed to conduct each acquisition session.

1. Place the NDI camera.
2. Place the global axis tool on the side of the treadmill.
3. Turn on the Polaris NDI camera and verify the visibility of the pelvis-attached markers and the global reference tool using the software NDI Track.
4. Place both Zed2i cameras on the same spots from the previous trials and set the preferred tripod height. It is important to have the camera placed on the same place each time for a reproducible camera perspective. One camera is placed in front of the person while the other one is placed on his right side. Due to some logistic limitations, we were not able to place a frontal camera in front of the person during the walking trials at Emovi. Some trials were also recorded before including the frontal camera setup in the process: only lateral view was available.
5. Connect both cameras to the computer and verify the visibility of the whole participant on both cameras.
6. Make sure the participant has is eligible for the KneeKG trial (no knee problems) and is wearing shorts to allow the KneeKG placement.
7. Take the participant's consent by signing the dedicated form.
8. The participant makes an unrecorded trial to set the treadmill speed to his comfortable walking speed
9. Place the appropriate KneeKG accessories on the tibia, the Knee and around the belt as shown in Figure 4.1.
10. The patient walks for 30 seconds to make sure the KneeKG is well placed and does not fall during the trial.
11. The patient is instructed to do some functional movements cited by the Knee3D^{TM} software for the anatomical calibration. This step allows to obtain the anatomical tibial and femur frames orientation with respect to the technical frames determined by the reflective markers.

12. The participant does two walking trials of approximately 45 seconds each.
13. The participant is instructed to make three static movements during the same trial: standing maximal extension stance, standing maximal flexion stance, and standing straight stance for 20 seconds each.
14. The participant is instructed to make a squat stance for 30 seconds.

BIBLIOGRAPHY

Andriluka, M., Pishchulin, L., Gehler, P. & Schiele, B. (2014, June). 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Antognini, C., Ortigas-Vásquez, A., Knowlton, C., Utz, M., Sauer, A. & Wimmer, M. A. (2025). Comparison of markerless and marker-based motion analysis accounting for differences in local reference frame orientation. *Journal of Biomechanics*, 112683. doi: 10.1016/j.jbiomech.2025.112683.

Baker, R. (2006). Gait analysis methods in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 3(1), 4. doi: 10.1186/1743-0003-3-4.

Belagiannis, V. & Zisserman, A. (2017). Recurrent Human Pose Estimation. *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 468-475. doi: 10.1109/FG.2017.64.

Bradshaw, T. J., Huemann, Z., Hu, J. & Rahmim, A. (2023). A Guide to Cross-Validation for Artificial Intelligence in Medical Imaging. *Radiology: Artificial Intelligence*, 5(4), e220232. doi: 10.1148/ryai.220232.

C. G. Lab. [[Online; accessed 27-May-2025]]. (2003). Carnegie-Mellon Motion Capture (MoCap) Database. Retrieved from: http://mocap.cs.cmu.edu/info.php.

Cai, N., Su, Z., Lin, Z., Wang, H., Yang, Z. & Ling, B. W.-K. (2017). Blind inpainting using the fully convolutional neural network. *The Visual Computer*, 33(2), 249–261. doi: 10.1007/s00371-015-1190-z.

Cao, Z., Simon, T., Wei, S.-E. & Sheikh, Y. (2017). Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1302-1310. doi: 10.1109/CVPR.2017.143.

Cao, Z., Chu, Z., Liu, D. & Chen, Y. (2021). A Vector-based Representation to Enhance Head Pose Estimation. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1187-1196. doi: 10.1109/WACV48630.2021.00123.

Cawley, G. C. & Talbot, N. L. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *J. Mach. Learn. Res.*, 11, 2079–2107.

Charles, J., Pfister, T., Everingham, M. & Zisserman, A. (2014). Automatic and Efficient Human Pose Estimation for Sign Language Videos. *Int. J. Comput. Vision*, 110(1), 70–90. doi: 10.1007/s11263-013-0672-6.

Chen, Y., Shen, C., Wei, X.-S., Liu, L. & Yang, J. (2017). Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1221-1230. doi: 10.1109/ICCV.2017.137.

Chen, Y., Tian, Y. & He, M. (2020). Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*, 192, 102897. doi: https://doi.org/10.1016/j.cviu.2019.102897.

Clément, J. (2015). *3D kinematic analysis of healthy and osteoarthritic knee during squat with multi-body optimisation: performance of subject-specific joint models*. (Ph.D. thesis).

De Filippi, E., Wolter, M., Melo, B. R. P., Tierra-Criollo, C. J., Bortolini, T., Deco, G. & Moll, J. (2021). Classification of Complex Emotions Using EEG and Virtual Environment: Proof of Concept and Therapeutic Implication. *Frontiers in Human Neuroscience*, Volume 15 - 2021. doi: 10.3389/fnhum.2021.711279.

Deakin, R. (1999). 3-D Coordinate Transformations. 58, 223-34.

Demir, U. & Ünal, G. B. (2018). Patch-Based Image Inpainting with Generative Adversarial Networks. *CoRR*, abs/1803.07422. Retrieved from: http://arxiv.org/abs/1803.07422.

Diebel, J. (2006). Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. *Matrix*, 58.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G. & Sun, J. (2021, June). RepVGG: Making VGG-style ConvNets Great Again . *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13728-13737. doi: 10.1109/CVPR46437.2021.01352.

Dinh, H., Le, S., Than, M., Ho, M., Vuilerrme, N. & Pham, H. (2025). Quantitative Gait Analysis from Single RGB Videos Using a Dual-Input Transformer-Based Network. *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*, pp. 1-5. doi: 10.1109/ISBI60581.2025.10981132.

Doosti, B., Naha, S., Mirbagheri, M. & Crandall, D. J. (2020, June). HOPE-Net: A Graph-Based Model for Hand-Object Pose Estimation . *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6607-6616. doi: 10.1109/CVPR42600.2020.00664.

Dubey, S. & Dixit, M. (2023). A comprehensive survey on human pose estimation approaches. *Multimedia Systems*, 29(1), 167–195. doi: 10.1007/s00530-022-00980-0.

Fanelli, G., Dantone, M., Gall, J., Fossati, A. & Van Gool, L. (2013). Random Forests for Real Time 3D Face Analysis. *Int. J. Comput. Vision*, 101(3), 437–458.

Fiorentino, N. M., Atkins, P. R., Kutschke, M. J., Goebel, J. M., Foreman, K. B. & Anderson, A. E. (2017). Soft tissue artifact causes significant errors in the calculation of joint angles and range of motion at the hip. *Gait & Posture*, 55, 184–190. doi: 10.1016/j.gaitpost.2017.03.033.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587. doi: 10.1109/CVPR.2014.81.

Glorot, X., Bordes, A. & Bengio, Y. (2011, April). Deep Sparse Rectifier Neural Networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 15(Proceedings of Machine Learning Research), 315–323. Retrieved from: https://proceedings.mlr.press/v15/glorot11a.html.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press.

Grassia, F. S. (1998). Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3), 29–48. doi: 10.1080/10867651.1998.10487493.

Hagemeister, N., Parent, G., Van De Putte, M., St-Onge, N., Duval, N. & De Guise, J. (2005). A reproducible method for studying three-dimensional knee kinematics. *Journal of Biomechanics*, 38(9), 1926–1931. doi: 10.1016/j.jbiomech.2005.05.013.

Hammerla, N. Y., Halloran, S. & Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, (IJCAI'16), 1533–1540.

Hannun, A., Rajpurkar, P., Haghpanahi, M., Tison, G., Bourn, C., Turakhia, M. & Ng, A. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25. doi: 10.1038/s41591-018-0268-3.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778. doi: 10.1109/CVPR.2016.90.

He, K., Gkioxari, G., Dollár, P. & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988. doi: 10.1109/ICCV.2017.322.

Hempel, T., Abdelrahman, A. A. & Al-Hamadi, A. (2022, October). 6d Rotation Representation For Unconstrained Head Pose Estimation. *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 2496–2500. doi: 10.1109/icip46576.2022.9897219.

Hsu, H. & Siwiec, R. M. (2023, June). Knee osteoarthritis. Retrieved from: https://www.ncbi.nlm.nih.gov/books/NBK507884/.

Huang, G. B., Mattar, M., Berg, T. & Learned-Miller, E. (2008, October). Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. Retrieved from: https://inria.hal.science/inria-00321923.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, (ICML'15), 448–456.

Ionescu, C., Papava, D., Olaru, V. & Sminchisescu, C. (2014). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1325–1339. doi: 10.1109/TPAMI.2013.248.

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J. & Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2), 203–211. doi: 10.1038/s41592-020-01008-z.

Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. (2017, July). Image-to-Image Translation with Conditional Adversarial Networks . *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967-5976. doi: 10.1109/CVPR.2017.632.

Jamsrandorj, A., Kumar, K. S., Arshad, M. Z., Mun, K.-R. & Kim, J. (2022). Deep Learning Networks for View-independent Knee and Elbow Joint Angle Estimation. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2703-2707. doi: 10.1109/EMBC48229.2022.9871106.

Johnson, J., Alahi, A. & Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *Computer Vision – ECCV 2016*, pp. 694–711.

Johnson, S. & Everingham, M. (2010). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. *British Machine Vision Conference*. Retrieved from: https://api.semanticscholar.org/CorpusID:7318714.

Josyula, R. & Ostadabbas, S. (2021). A Review on Human Pose Estimation. *CoRR*, abs/2110.06877. Retrieved from: https://arxiv.org/abs/2110.06877.

Kadaba, M. P., Ramakrishnan, H. K. & Wootten, M. E. (1990). Measurement of lower extremity kinematics during level walking. *Journal of Orthopaedic Research®*, 8(3), 383–392. doi: 10.1002/jor.1100080310.

Kapoor, S. & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 100804. doi: https://doi.org/10.1016/j.patter.2023.100804.

Ke, L., Chang, M.-C., Qi, H. & Lyu, S. (2018, September). Multi-Scale Structure-Aware Network for Human Pose Estimation. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Kharb, A., Saini, V., Jain, Y., Dhiman, S., Tech, M. & Scholar. (2011). *IJCEM Int J Comput Eng Manag*, 13.

Kidzinski, L., Yang, B., Hicks, J. L., Rajagopal, A., Delp, S. L. & Schwartz, M. H. (2020). Deep neural networks enable quantitative movement analysis using single-camera videos. *Nature Communications*, 11(1), 4054. doi: 10.1038/s41467-020-17807-z.

Knap, P. (2024). Human Modelling and Pose Estimation Overview. Retrieved from: https://arxiv.org/abs/2406.19290.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, (IJCAI'95), 1137–1143.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25. Retrieved from: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Krstajic, D., Buturovic, L. J., Leahy, D. E. & Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, 6(1). doi: 10.1186/1758-2946-6-10.

Kuhn, M. [R package version 1.3-1]. (2012). QSARdata: Data Sets for QSAR. Retrieved from: https://CRAN.R-project.org/package=QSARdata.

Kuipers, J. B. (1999). *Quaternions and rotation sequences*. Princeton University Pres. doi: 10.1515/9780691211701.

Kumar, K. S., Jamsarndorj, A., Jung, D., Lee, D., Kim, J. & Mun, K.-R. (2021). Vision-based human joint angular velocity estimation during squat and walking on a treadmill actions. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2186-2190. doi: 10.1109/EMBC46164.2021.9630438.

Le, H. & Pham, H. (2024). Learning to Estimate Critical Gait Parameters from Single-View RGB Videos with Transformer-Based Attention Network. *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1-5. doi: 10.1109/ISBI56570.2024.10635218.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z. & Tu, Z. (2015, May). Deeply-Supervised Nets. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 38(Proceedings of Machine Learning Research), 562–570. Retrieved from: https://proceedings.mlr.press/v38/lee15a.html.

Li, K., Wang, S., Zhang, X., Xu, Y., Xu, W. & Tu, Z. (2021, June). Pose Recognition with Cascade Transformers . *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1944-1953. doi: 10.1109/CVPR46437.2021.00198.

LI, S., Liu, Z.-Q. & Chan, A. B. (2014). Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 488-495. doi: 10.1109/CVPRW.2014.78.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *Computer Vision – ECCV 2014*, pp. 740–755.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Laak, J. A. W. M. v. d., Ginneken, B. v. & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. doi: https://doi.org/10.1016/j.media.2017.07.005.

Lombardi, A., Diacono, D., Amoroso, N., Monaco, A., Tavares, J. M. R. S., Bellotti, R. & Tangaro, S. (2021). Explainable Deep Learning for Personalized Age Prediction With Brain Morphology. *Frontiers in Neuroscience*, Volume 15 - 2021. doi: 10.3389/fnins.2021.674055.

Lombardi, A., Diacono, D., Amoroso, N., Biecek, P., Monaco, A., Bellantuono, L., Pantaleo, E., Logroscino, G., De Blasi, R., Tangaro, S. & Bellotti, R. (2022). A robust framework to investigate the reliability and stability of explainable artificial intelligence markers of Mild Cognitive Impairment and Alzheimer's Disease. *Brain Informatics*, 9(1). doi: 10.1186/s40708-022-00165-5.

Lustig, S., Magnussen, R. A., Cheze, L. & Neyret, P. (2012). The KneeKG system: a review of the literature. *Knee Surgery Sports Traumatology Arthroscopy*, 20(4), 633–638. doi: 10.1007/s00167-011-1867-4.

Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S. & Chen, K. (2022). RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *ArXiv*, abs/2212.07784. Retrieved from: https://api.semanticscholar.org/CorpusID:254685870.

Maas, A. L. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. Retrieved from: https://api.semanticscholar.org/CorpusID:16489696.

Martin Bland, J. & Altman, D. (1986). STATISTICAL METHODS FOR ASSESSING AGREEMENT BETWEEN TWO METHODS OF CLINICAL MEASUREMENT. *The Lancet*, 327(8476), 307–310. doi: 10.1016/S0140-6736(86)90837-8. Publisher: Elsevier.

Martinez, J., Hossain, R., Romero, J. & Little, J. J. (2017). A Simple Yet Effective Baseline for 3d Human Pose Estimation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2659-2668. doi: 10.1109/ICCV.2017.288.

McCambridge, J., Witton, J. & Elbourne, D. R. (2013). Systematic review of the Hawthorne effect: New concepts are needed to study research participation effects. *Journal of Clinical Epidemiology*, 67(3), 267–277. doi: 10.1016/j.jclinepi.2013.08.015.

McGinley, J. L., Baker, R., Wolfe, R. & Morris, M. E. (2009). The reliability of three-dimensional kinematic gait measurements: A systematic review. *Gait & Posture*, 29(3), 360-369. doi: https://doi.org/10.1016/j.gaitpost.2008.09.003.

McIntosh, K. (2019). Computing Euler angles from a rotation matrix. *rpi*. Retrieved from: https://www.academia.edu/39656139/Computing_Euler_angles_from_a_rotation_matrix.

Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W. & Theobalt, C. (2017). Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. *2017 International Conference on 3D Vision (3DV)*, pp. 506-516. doi: 10.1109/3DV.2017.00064.

Mentiplay, B. F., Perraton, L. G., Bower, K. J., Pua, Y.-H., McGaw, R., Heywood, S. & Clark, R. A. (2015). Gait assessment using the Microsoft Xbox One Kinect: Concurrent validity and inter-day reliability of spatiotemporal and kinematic variables. *Journal of Biomechanics*, 48(10), 2166–2170. doi: 10.1016/j.jbiomech.2015.05.021.

Na, A. & Buchanan, T. S. (2019). Self-reported walking difficulty and knee osteoarthritis influences limb dynamics and muscle co-contraction during gait. *Human Movement Science*, 64, 409-419. doi: https://doi.org/10.1016/j.humov.2018.11.002.

Newell, A., Yang, K. & Deng, J. (2016). Stacked Hourglass Networks for Human Pose Estimation. *Computer Vision – ECCV 2016*, pp. 483–499.

Owens, N., Harris, C. & Stennett, C. (2003). Hawk-eye tennis system. *2003 International Conference on Visual Information Engineering VIE 2003*, pp. 182-185. doi: 10.1049/cp:20030517.

Pataky, T. (2011). One-dimensional statistical parametric mapping in Python. *Computer methods in biomechanics and biomedical engineering*, 15, 295-301. doi: 10.1080/10255842.2010.527837.

Pataky, T. C. (2010). Generalized n-dimensional biomechanical field analysis using statistical parametric mapping. *Journal of Biomechanics*, 43(10), 1976–1982. doi: 10.1016/j.jbiomech.2010.03.008.

Pavllo, D., Feichtenhofer, C., Grangier, D. & Auli, M. (2019). 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7745-7754. doi: 10.1109/CVPR.2019.00794.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. & Louppe, G. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12.

Peretroukhin, V., Giamou, M., Greene, W. N., Rosen, D., Kelly, J. & Roy, N. (2020, July). A Smooth Representation of Belief over SO(3) for Deep Rotation Learning with Uncertainty. *Proceedings of Robotics: Science and Systems*. doi: 10.15607/RSS.2020.XVI.007.

Perry, J. & Burnfield, J. M. (2010). *Gait Analysis: Normal and Pathological Function* (ed. 2). Thorofare, NJ: SLACK Incorporated.

Petersen, R. C., Aisen, P. S., Beckett, L. A., Donohue, M. C., Gamst, A. C., Harvey, D. J., Jack, C. R., Jagust, W. J., Shaw, L. M., Toga, A. W., Trojanowski, J. Q. & Weiner, M. W. (2010). Alzheimer's Disease Neuroimaging Initiative (ADNI). *Neurology*, 74(3), 201-209. doi: 10.1212/WNL.0b013e3181cb3e25.

Quan, W., Chen, J., Liu, Y., Yan, D.-M. & Wonka, P. (2024). Deep Learning-Based Image and Video Inpainting: A Survey. *International Journal of Computer Vision*, 132(7), 2367–2400. doi: 10.1007/s11263-023-01977-6.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D. Y., Bagul, A., Langlotz, C. P., Shpanskaya, K. S., Lungren, M. P. & Ng, A. Y. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *CoRR*, abs/1711.05225. Retrieved from: http://arxiv.org/abs/1711.05225.

Refaeilzadeh, P., Tang, L. & Liu, H. (2009). Cross-Validation. In LIU, L. & ÖZSU, M. T. (Eds.), *Encyclopedia of Database Systems* (pp. 532–538). Boston, MA: Springer US. doi: 10.1007/978-0-387-39940-9_565.

Ren, S., He, K., Girshick, R. & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(06), 1137-1149. doi: 10.1109/TPAMI.2016.2577031.

Rodrigues, T. B., Catháin, C. O., Devine, D., Moran, K., O'Connor, N. E. & Murray, N. (2019). An evaluation of a 3D multimodal marker-less motion analysis system. *Proceedings of the 10th ACM Multimedia Systems Conference*, (MMSys '19), 213–221. doi: 10.1145/3304109.3306236.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211-252. doi: 10.1007/s11263-015-0816-y.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S. & Pantic, M. (2013). 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. *2013 IEEE International Conference on Computer Vision Workshops*, pp. 397-403. doi: 10.1109/ICCVW.2013.59.

Salem, N. (2021). A Survey on Various Image Inpainting Techniques. *Future Engineering Journal*, 2. doi: 10.54623/fue.fej.2.2.1.

Salem, N., Mahdi, H. & Abbas, H. (2019). Random-Shaped Image Inpainting using Dilated Convolution. *International Journal of Engineering and Advanced Technology*, 8, 641 - 647. doi: 10.35940/ijeat.F8089.088619.

Sapp, B. & Taskar, B. (2013). MODEC: Multimodal Decomposable Models for Human Pose Estimation. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674-3681. doi: 10.1109/CVPR.2013.471.

Shorten, C. & Khoshgoftaar, T. M. (2019a). A survey on Image Data Augmentation for Deep Learning. *Journal Of Big Data*, 6(1). doi: 10.1186/s40537-019-0197-0.

Shorten, C. & Khoshgoftaar, T. M. (2019b). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. doi: 10.1186/s40537-019-0197-0.

Sigal, L., Balan, A. & Black, M. (2010). HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision*, 87, 4-27. doi: 10.1007/s11263-009-0273-6.

Stenum, J., Rossi, C. & Roemmich, R. T. (2021). Two-dimensional video-based analysis of human gait using pose estimation. *PLoS Computational Biology*, 17(4), e1008935. doi: 10.1371/journal.pcbi.1008935.

Sun, K., Xiao, B., Liu, D. & Wang, J. (2019a). Deep High-Resolution Representation Learning for Human Pose Estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5686-5696. doi: 10.1109/CVPR.2019.00584.

Sun, T., Fang, W., Chen, W., Yao, Y., Bi, F. & Wu, B. (2019b). High-Resolution Image Inpainting Based on Multi-Scale Neural Network. *Electronics*, 8, 1370. doi: 10.3390/electronics8111370.

Sun, X., Shang, J., Liang, S. & Wei, Y. (2017). Compositional Human Pose Regression. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2621-2630. doi: 10.1109/ICCV.2017.284.

Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K. & Lempitsky, V. (2022, January). Resolution-Robust Large Mask Inpainting With Fourier Convolutions. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2149-2159.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9. doi: 10.1109/CVPR.2015.7298594.

Tompson, J., Jain, A., LeCun, Y. & Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, (NIPS'14), 1799–1807.

Toshev, A. & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653-1660. doi: 10.1109/CVPR.2014.214.

Tripathi, S., Ranade, S., Tyagi, A. & Agrawal, A. (2020). PoseNet3D: Learning Temporally Consistent 3D Human Pose via Knowledge Distillation. *2020 International Conference on 3D Vision (3DV)*, pp. 311-321. doi: 10.1109/3DV50981.2020.00041.

Wade, L., Needham, L., McGuigan, P. & Bilzon, J. (2022). Applications and limitations of current markerless motion capture methods for clinical gait biomechanics. *PeerJ*, 10, e12995. doi: 10.7717/peerj.12995.

Wainer, J. & Cawley, G. (2021). Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, 182, 115222. doi: https://doi.org/10.1016/j.eswa.2021.115222.

Wang, C., Zhang, F. & Ge, S. S. (2021). A comprehensive survey on 2D multi-person pose estimation methods. *Engineering Applications of Artificial Intelligence*, 102, 104260. doi: https://doi.org/10.1016/j.engappai.2021.104260.

Wang, P., Chen, Y., Su, W., Wang, J., Ma, T. & Yu, H. (2024). Beyond Gait: Learning Knee Angle for Seamless Prosthesis Control in Multiple Scenarios. Retrieved from: https://arxiv.org/abs/2404.06772.

Wei, S.-E., Ramakrishna, V., Kanade, T. & Sheikh, Y. (2016). Convolutional Pose Machines. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724-4732. doi: 10.1109/CVPR.2016.511.

White, J. & Power, S. D. (2023). K-Fold Cross-Validation can significantly Over-Estimate true classification accuracy in common EEG-Based passive BCI Experimental Designs: An Empirical investigation. *Sensors*, 23(13), 6077. doi: 10.3390/s23136077.

Winter, D. A. (2009). *Biomechanics and motor control of human movement*. Wiley. doi: 10.1002/9780470549148.

Xiao, B., Wu, H. & Wei, Y. (2018, September). Simple Baselines for Human Pose Estimation and Tracking. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Xu, Z., Zhang, X., Chen, W., Yao, M., Liu, J., Xu, T. & Wang, Z. (2023). A review of image inpainting methods based on deep learning. *Applied Sciences*, 13(20), 11189. doi: 10.3390/app132011189.

Yang, T.-Y., Chen, Y.-T., Lin, Y.-Y. & Chuang, Y.-Y. (2019). FSA-Net: Learning Fine-Grained Structure Aggregation for Head Pose Estimation From a Single Image. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1087-1096. doi: 10.1109/CVPR.2019.00118.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. & Huang, T. (2019). Free-Form Image Inpainting With Gated Convolution. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4470-4479. doi: 10.1109/ICCV.2019.00457.

Zagoruyko, S. & Komodakis, N. (2017). Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. Retrieved from: https://openreview.net/forum?id=Sks9_ajex.

Zeiler, M. D. & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *Computer Vision – ECCV 2014*, pp. 818–833.

Zeng, Y., Fu, J., Chao, H. & Guo, B. (2019). Learning Pyramid-Context Encoder Network for High-Quality Image Inpainting. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1486-1494. doi: 10.1109/CVPR.2019.00158.

Zhang, H., Wang, M., Liu, Y. & Yuan, Y. (2020). FDN: Feature Decoupling Network for Head Pose Estimation. *AAAI Conference on Artificial Intelligence*. Retrieved from: https://api.semanticscholar.org/CorpusID:213818512.

Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N. & Shah, M. (2023). Deep Learning-based Human Pose Estimation: A Survey. *ACM Comput. Surv.*, 56(1). doi: 10.1145/3603618.

Zhou, Y., Barnes, C., Lu, J., Yang, J. & Li, H. (2019). On the Continuity of Rotation Representations in Neural Networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5738-5746. doi: 10.1109/CVPR.2019.00589.

Zhou, Y. & Gregson, J. (2020). WHENet: Real-time Fine-Grained Estimation for Wide Range Head Pose. *ArXiv*, abs/2005.10353. Retrieved from: https://api.semanticscholar.org/CorpusID:218763523.

Zong-Hao,MA, C., Li, Z. & He, C. (2023). Advances in Biomechanics-Based motion analysis. *Bioengineering*, 10(6), 677. doi: 10.3390/bioengineering10060677.