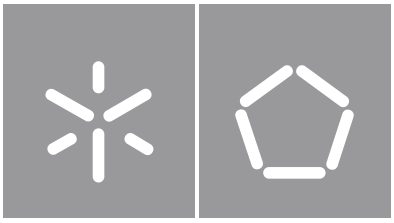
Universidade do Minho
Escola de Engenharia

Carolina Maria Cunha Gonçalves

A Deep Learning solution for real-time human motion decoding in smart walkers

março de 2022

A Deep Learning solution for real-time human motion decoding in smart walkers
Carolina Maria Cunha Gonçalves
UMinho | 2022



Universidade do Minho
Escola de Engenharia

Carolina Maria Cunha Gonçalves

A Deep Learning solution for real-time human motion decoding in smart walkers

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica
Ramo de Electrónica Médica

Trabalho efetuado sob a orientação de
Professora Doutora Cristina P. Santos
Professora Doutora Sara Moccia

março de 2022

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Acknowledgments

The project developed throughout this year could not be possible without the help and encouragement of family, co-workers and friends.

Firstly, I want to thank my family, parents and sister, for the provided conditions, healthy environment and support, during all my academic journey. And, particularly to my sister, an appreciation for all the medicine related answered questions and discussions.

I would also like to express my truthful gratitude to my adviser Prof. Cristina Santos, for all the guidance, motivation and dedication, fostering multiple opportunities and projects to explore and find my real interests, as well as valuable research connections that helped me to grow and expand my knowledge. This work wouldn't be possible without her contributions and research suggestions.

I am grateful to all the VRAI laboratory (Università Politecnica delle Marche) for hosting me, with a special thank you to my co-adviser Sara Moccia for mentoring and teaching me, as well as to her co-workers Daniele Berardini and Lucia Migliorelli for all their availability.

Thanks also to my BIRDLAB colleagues for the great learning environment, always with good disposition and knowledge to be shared. I'm thankful for all the help and debated ideas, specially to João Lopes, for all the hard working, motivation, support and a little bit of patience.

Statement of Integrity

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Abstract

The treatment of gait impairments has increasingly relied on rehabilitation therapies which benefit from the use of smart walkers. These walkers still lack advanced and seamless Human-Robot Interaction, which intuitively understands the intentions of human motion, empowering the user's recovery state and autonomy, while reducing the physician's effort.

This dissertation proposes the development of a deep learning solution to tackle the human motion decoding problematic in smart walkers, using only lower body vision information from a camera stream, mounted on the WALKit Smart Walker, a smart walker prototype for rehabilitation purposes.

Different deep learning frameworks were designed for early human motion recognition and detection. A custom acquisition method, including a smart walker's automatic driving algorithm and labelling procedure, was also designed to enable further training and evaluation of the proposed frameworks.

Facing a 4-class (stop, walk, turn right/left) classification problem, a deep learning convolutional model with an attention mechanism achieved the best results: an offline f1-score of 99.61%, an online calibrated instantaneous precision higher than 97% and a human-centred focus slightly higher than 30%.

Promising results were attained for early human motion detection, with enhancements in the focus of the proposed architectures. However, further improvements are still needed to achieve a more reliable solution for integration in a smart walker's control strategy, based in the human motion intentions.

Keywords: Computer vision, deep learning, early action detection, early action recognition, human motion decoding, Human-Robot Interaction, smart walkers

Resumo

O tratamento de distúrbios da marcha tem apostado cada vez mais em terapias de reabilitação que beneficiam do uso de andarilhos inteligentes. Estes ainda carecem de uma Interação Humano-Robô avançada e eficaz, capaz de entender, intuitivamente, as intenções do movimento humano, fortalecendo a recuperação autónoma do paciente e reduzindo o esforço médico.

Esta dissertação propõe o desenvolvimento de uma solução de aprendizagem para o problema de descodificação de movimento humano em andarilhos inteligentes, usando apenas vídeos recolhidos pelo *WALKit Smart Walker*, um protótipo de andarilho inteligente usado para reabilitação.

Foram desenvolvidos algoritmos de aprendizagem para o reconhecimento e detecção precoces de movimento humano. Um método de aquisição personalizado, incluindo um algoritmo de condução e labelização automatizados, foi projetado para permitir o conseguinte treino e avaliação dos algoritmos propostos.

Perante a classificação de 4 ações (parar, andar, virar à direita/esquerda), um modelo convolucional com um mecanismo de atenção alcançou os melhores resultados: f1-score *offline* de 99,61%, precisão instantânea calibrada online de superior a 97 % e um foco centrado no ser humano ligeiramente superior a 30%.

Com esta dissertação alcançaram-se resultados promissores para a detecção precoce de movimento humano, com aprimoramentos no foco dos algoritmos propostos. No entanto, ainda são necessárias melhorias adicionais para alcançar uma solução mais robusta para a integração na estratégia de controlo de um andarilho inteligente, com base nas intenções de movimento do utilizador.

Palavras-Chave: Andarilhos inteligentes, aprendizagem profunda, descodificação de movimento humano, deteção precoce da ação, Interação Humano-Robô, reconhecimento precoce de ação, visão computador

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 3, 5, 10, 13, 15

cIP instantaneous calibrated precision. 8, 58, 92

CNN Convolutional Neural Network. 19–26, 39, 47–51, 59, 84, 85, 89

COM Center-of-Mass. 13, 14

CV Computer Vision. 3, 10, 13, 15–17, 19, 26, 29

DL Deep learning. xiii, xiv, 1, 3–10, 13–17, 19–21, 26, 27, 29, 37, 47, 52, 55, 79, 84, 91–93, 95

ERF Effective Receptive Field. 42

FC Fully Connected. 14, 21, 22

FN False Negatives. 23, 57, 58, 82

FP False Positives. 23, 57, 58, 82, 83, 94

GAP Global Average Pooling. 49–51

GT Ground-Truth. xiii, xv, 9, 51, 58, 59, 64, 66, 68–70, 73–77, 80–84, 86–88, 91–93

HAP Human Action Prediction. xiv, 15, 17, 20, 22, 26, 47

HAR Human Action Recognition. xiv, 15–18, 20, 26, 49

HRI Human-Robot Interaction. 1–4, 10, 15, 26, 30, 35

IA instantaneous accuracy. 5, 8, 58, 87

IMU Inertial Measurement Unit. 3, 4, 11, 12, 25, 28, 30, 32

IP instantaneous precision. 5, 8, 58, 87

IR infrared. 3

LSTM Long Short-Term Memory. 14, 21–26, 84, 89

MI Motion Intention. 3, 4, 8–13, 15, 25, 26, 33, 88, 91

MLP Multi Layer Perceptron. 23

NN Neural Network. 12, 21, 30, 42

OAD Online Action Detection. 15, 16, 21, 54, 58

OF Optical Flow. 18, 19, 38, 39

PID Proportional, Integral and Derivative. 13, 29, 36, 95, 96

ReLU Rectified Linear Unit. 49

RL Reinforcement learning. 14

RNN Recurrent Neural Network. 20, 23, 39, 93

ROI Region of Interest. 41, 44, 45, 81, 82, 93

ROS Robot Operating System. 29, 35

SW Smart Walker. 1–6, 8–13, 25–30, 33–39, 42, 44, 47, 54, 80, 83, 87–89, 91, 92, 94, 95

TL Turn Left. xiii, 7, 25, 34, 65, 74, 87, 88, 91

TN True Negatives. 57, 58

TP True Positives. 57, 58, 65, 72, 82, 83

TR Turn Right. xiii, 7, 25, 34, 73, 74, 88, 91

wIA instantaneous weighted accuracy. 8, 58, 92

Chapter 1

Introduction

This dissertation presents the work carried out over the past year, integrated in the scope of the Master's Degree in Biomedical Engineering at the Biomedical Robotic Devices Lab (BIRDLAB) included in the Center for MicroElectroMechanical Systems (CMEMS), a research center of the Department of Industrial Electronics (DEI) of University of Minho. This project was developed in colaboration with the Vision Robotics Artificial Intelligence (VRAI) laboratory, in the Università Politecnica delle Marche.

This project main goal was to deploy an algorithm capable of inferring the human walking intentions, such as stop, walking straight and turning right or left, from RGB streams recorded with camera embedded in the WALKit Smart Walker (SW), pointed to the legs and feet, used in patient rehabilitation in hospital environments. The approach aims to be integrated in a future control strategy able to drive the device, according to the human intents, and so fostering a seamless Human-Robot Interaction (HRI).

This project was divided in three main phases: **i)** elaboration of data acquisition procedures, along with their execution, to create a dataset for human motion decoding; **ii)** research and development of several vision-based DL approaches to early detect distinct walking events, using the collected data, and to enhance the extraction of relevant features; **iii)** offline evaluation of these approaches, as well as further selection of the most suitable algorithm for real-time applications in a WALKit Smart Walker control strategy, including real-time simulations to evaluate the latter's performance.

1.1 Motivation

Disability is a prominent part of human condition. According to the World Health Organisation, in the year of 2018, over a billion people, about 15% of the world's population, were estimated to live with some

form of disability [1]. This number is still rising, due, not only to the ageing population, but also to an increase in chronic health conditions [1]. Across Europe, for example, the dysfunctional gait is seen as the most frequent form of disability, where 5 million citizens are estimated to depend on a wheelchair [2]. Gait and posture impairments are present in almost all neurodegenerative disorders, such as tumours, aneurysms, cerebellar ataxia, cerebral palsy, strokes, Parkinson's disease and others, and can have severe repercussions [2]–[5]. This kind of pathology can trigger permanent changes in strength, sensation and movement, as well as other body functions, leading to lack of stability and increased risk of falls and fall-related morbidity [2] [6]. Consequently, the patients suffer from a **reduced independence and quality of life,** with associated social-economic consequences due to the **increased institutionalisation and dependence on others** [5]–[7]. Disabilities imply several economic and social costs to individuals, families, communities and nations [8]. For instance, in 2010, it was estimated an European total annual cost of 64 billion for stroke and 14 billion for Parkinson's disease [7]. Therefore, **there is an emerging need for developing mobility assistive methods and technologies** that can support and contribute to the rehabilitation of the elderly and other impaired populations.

Rehabilitation therapies already revealed promising results tackling these impairments [9]. However, they imply a high burden of care to clinicians [2] and a dependency over clinicians' experience [8]. Gait rehabilitation requires long periods of intense physical exercise, presenting challenges for physiotherapists, due to the high demand of physical effort, plus the intra- and inter- clinician and patients variability, which turns this kind of therapy into a more time consuming one and prone to errors [2]. In order to overcome these obstacles, assistive technologies have emerged as effective means to increase subject's independence and participation in their rehabilitation therapies [8]. Recent technological advances have culminated in a new type of intelligent assistive mobility devices, the so-called "smart walkers" (SW), "robotic walkers" or "robotic rollators". Such robotic systems are intended to be used by or with humans, implying the understanding, designing and evaluation of the device towards a safe and efficient robot-human interaction. Therefore, current assistive devices no longer serve as just a conventional physical supporter, but comprehend now other intelligent functionalities that target the development of a robust HRI. The deployed functionalities have been focusing on gait assistance, obstacle avoidance, navigation assistance, sit-to-stand transfer, body weight support or fall prevention [10] [11]. Nonetheless, further enhancements and interaction goals still need to be tackled, in order to develop and integrate a more advanced HRI, capable of empowering the user's recovery state. For instance, the SW should be able to, not only provide aid and analyse multi-sensory signals related to gait and posture features, but also understand human actions, intentions and needs. Particularly, a method for **natural and intuitive**

adaptation to the user's needs, with minimal interference, is essential for **seamless HRI**. A device capable of predicting human walking intentions as a driving control strategy would enable a more natural and anticipated assistance, encouraging the patient to take an active role in his own rehabilitation exercises or therapy sessions [12]. Such interaction should result from the device's built-in sensors, in order to maximise intuitiveness and technology acceptance. Inspired by humans, who can unconsciously predict how other people move around them, only by observation, research has been done, in the Artificial Intelligence (AI) field, aiming to mimic this **ability of using visual stimuli to forecast the human motion** (*e.g.* [13], [14]). This is still a developing area which is expected to be promising in human-in-the-loop control approaches for SW [15].

1.2 Problem Statement

Human intention detection in SW prototypes has already been explored to provide commands to control the robot's velocity and position. Usually, human intention detection is achieved through the use of wearable sensors, such as Inertial Measurement Unit (IMU) [16], or embedded ones, like force sensors [17] [18] on the SW's handles or forearms support, Hall sensors [19], infrared (IR) sensors [20] or even lasers [12] [16] [21] [22].

Nevertheless, the usage of these sensory techniques presents some limitations and disadvantages, such as: increased number of sensors, whenever resorting to currently non-embedded sensors; demanding of set ups on the user's body (IMU), which **increases the duration of the rehabilitation sessions and adds an extra task for the physiotherapist to perform**; additional **cognitive effort** to correctly control the walker (force and hall sensors); **signal corruption** by realistic light conditions (IR sensors) or even by the electromagnetic interference of the walker's motors (IMU), which may imply **filtering processes**.

Some of the recently developed SWs incorporate cameras to attain multiple functionalities, but few exploit their potential for action recognition and human Motion Intention (MI) decoding. With this scope, traditional Computer Vision (CV) (*e.g.* [23] [24]) and DL algorithms (*e.g.* [25] [26]) have already been implemented, aiming a more intuitive, advanced and autonomous way to follow the user's intentions, without complex set ups of multiple sensors. However, there are still some open questions, specially considering the detection of the motion intention for control purposes [24]. Moreover, the studies presented so far still reveal premature results, not tested with a considerable sized samples of subjects and, most of the time, not really focused on walking directions (i.e., walk straight, turn left, turn right or stop), but on a more

divergent and wider spectrum of actions (e.g., sit, stand, walk, among others).

WALKit SW [27] is the robotic platform used throughout the development of this dissertation. This assistive device was specially designed to provide a personalised and user-oriented therapy, acting as a rehabilitation tool, while fostering HRI with active user participation. Currently, WALKit SW has a HRI solution implemented to decode the user's MI based on a specially designed handlebar embedded with an IMU. The HRI solution interprets the intention commands through heuristic rules, in order to classify the signals and control the motors' speed [27]. The user manipulates the handlebar to encode his/hers intentions and consequently drive the SW. This strategy could act as a dual-task therapy for patients already capable of performing manual control. Although it may be beneficial, the entailed manipulation could also translate into increased cognitive effort and perhaps divert the user's attention in therapy. For patients in early stages of rehabilitation, the walker can also be driven by remote control, removing these additional burden and/or distractions. However, the latter takes away the user's autonomy and control over his/hers rehabilitation. Therefore, it becomes important to create a more **intuitive, simple and technological advanced solution** to detect intentions and enable motor control to follow the human motion. This solution should be able to support the rehabilitation process and allow an active human involvement, while automatically adjusting the SWs direction and, in future, its velocity. This functionality would be suitable for later stages of rehabilitation, providing numerous advantages, namely: **i)** prompting the user's autonomy and complete focus on gait performance, by removing excessive aids or distractions (e.g. being driven/driving the walker); **ii)** tackling more specific rehabilitation aspects, in order to improve more complex tasks, like torso posture and foot positioning; and **iii)** reduce the clinician's effort, favouring the task of gait analysis, correction and assistance.

This dissertation aimed to address this challenge by proposing a DL-based strategy to decode human movement. This strategy used only the lower camera of WALKit SW, pointing at the legs and feet, which we hypothesise to be relevant to decode human movement and later control the device. The challenges are manifold since the deployment of this system implies a large amount of data to train the models and the ultimate model should be robust enough to deal with realistic environments, where challenging light conditions and feet occlusions may occur, compromising the input images. Once this is accomplished, this strategy shall be combined with an autonomous driving mode in order to provide more safety for patients, thus leading to a shared control strategy.

1.3 Goals

The ultimate goal of this dissertation was to develop an efficient non-invasive DL-based method able to directly decode human motion from RGB inputs. The developed solution has to be suitable for real-time performance, towards an human-in-the-loop control strategy. This work demanded dealing with the existent hardware, code development to control the SW prototype for data acquisition, as well as protocol designing, model training and testing in real-time simulations.

To reach this main objective, the following step-goals need to be achieved:

- **Goal 1:** To gather knowledge on the strategies implemented by SWs for human motion decoding through literature reviews, with special attention to the recent ones applying AI methods, and on the DL techniques used for video analysis, mainly aiming action recognition or future action prediction. A brief summary of these reviews will be presented on Sections 2.1 and 2.3, respectively.

- **Goal 2:** Devise a custom acquisition method, that enables the acquisition of multi-modal temporally synchronised data, for motion decoding purposes. This mimics realistic utilisation conditions and walking trajectories, as well as natural gaits and motion intentions, while providing automatic labels. All the executed procedures that enabled the process of data acquisition will be resumed in Section 3.2.

- **Goal 3:** Create a custom dataset with data acquired from multiple healthy subjects using the WALKit Smart Walker, in order to train and evaluate the different DL frameworks. This dataset can be used to create a balanced datasets of frames, as described in Section 3.3.

- **Goal 4:** Exploit DL frameworks, including tailored inputs design and post-processing techniques. The explored approaches should lead to, at least, 95% of accuracy for early recognition of human motions, whilst the majority of features have to be extracted within the human body region (with a minimum Mean Dice of 55%). The approaches that have been implemented, along with the post-processing and models' architectures, will be described in Chapter 4 and further details about input computation will be given in Section 3.4.1. Chapter 5 will reveal the obtained results.

- **Goal 5:** Evaluate the best framework in real-time simulations, mimicking, as much as possible, the conditions of its purposed future application. The model should be able to early detect the actions, with a minimum average of instantaneous accuracy (IA) and instantaneous precision (IP) of 95% and a maximum delay below 0.64s, which corresponds to the medium duration of one healthy

step, while walking at the fastest velocity assumed by the WALKit SW (1m/s) [1]. The evaluation results are shown in Section 5.3 and the key-performance indicators explained in Section 4.5.3.

1.4 Research Questions

Considering the ultimate goal of this dissertation and the step-goals presented, relevant research questions were identified, as follows:

- **RQ 1:** How to acquire data with the SW, without significantly disturbing the subject's gait and with a sufficiently accurate automatic labelling procedure? This question relates to **Goal 2** and it is answered in Section 3.2.
- **RQ 2:** Which inputs can be applied to the DL models that entail a low computational load, while encoding the human motion? This question relates to **Goal 4** and the answers will be found throughout Section 5.1.1.
- **RQ 3:** How can one improve the model's focus, leading it to mainly extract relevant features from the input's human body region? This question relates to **Goal 4** and the answers will be found in Sections 5.1.1, 5.1.2 and 5.2.
- **RQ 4:** Which DL model produces best results on early detecting the human motion considering a small window of the action? This question relates to **Goals 4** and **5** and is answered throughout Chapter 5.
- **RQ 5:** How effective and robust is the proposed DL solution for real-time applications towards a future human-in-the-loop control strategy? This final query is associated with **Goal 5** and answered in Section 5.3.

1.5 Solution Overview

Inspired on [14], which attempts to exploit action-aware features, this project proposes the use of visual information from the lower RGB-D camera embedded on the WALKit SW (Section 3.1) to generate different forms of input that encode human motion. This work was developed towards the final aimed solution, illustrated in Figure 1.1, aspiring the future integration of the devised DL solution in a human-in-the-loop control strategy.

[1] This step time was determined in laboratory experiments and it is in accordance with [28]

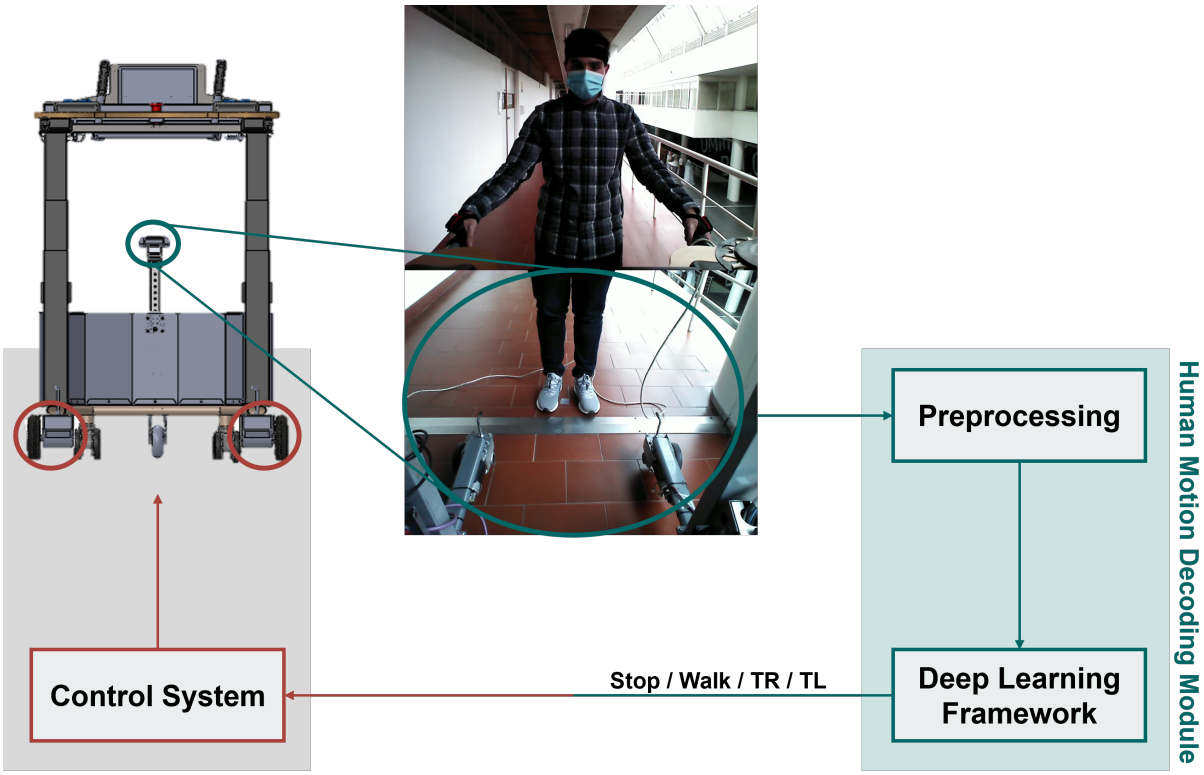


Figure 1.1: Overview of the proposed solution and final scope.

For that, a multi-modal dataset of multiple healthy subjects walking with the robotic device in real-world environments was acquired, considering an algorithm that allowed the generation of automatic trajectories. A sliding window was applied over the recorded videos at 30Hz, with a length of 4 and a stride of 2 frames. The resulting 4-frame sequences were converted to custom single-frame RGB inputs which, compared to the traditional temporal RGB sequences used in video analysis [14][29][30], demand less complex architectures. Different DL frameworks were trained and tested for early action recognition, while constantly evaluating and aiming to improve the relevance of the extracted features, so their focus would be mainly directed towards human legs and feet. These were considered here the action-aware features inherent to the target actions: STOP, WALK, TR and TL. Given the cyclic movement of walking, transfer-learning techniques were implemented in resemblance to [29], namely the use of pre-trained convolutional models, to prevent the model's overfitting to the acquired data.

To evaluate the model's performance, standard metrics of literature were used [29][14], namely accuracy, precision, recall and f1-score. Regarding the model's focus, this was evaluated through generated grad-CAMs [31], quantitatively compared with computed masks that segment the human body and whose algorithm was optimised in this dissertation.

Considering the classification results and the quality of the obtained grad-CAMs, a final model was

chosen and evaluated in real-time simulations, resorting to state-of-the-art online metrics (IA, instantaneous weighted accuracy (wIA), IP and instantaneous calibrated precision (cIP)) [32]. Post-processing techniques were also studied and implemented, to handle the model's uncertainty and improve the consistency of its output. The inference time of the whole approach, as well as the prediction delays, were determined to assess the applicability of this strategy in real-time control of the SW.

The process behind this solution is described in Figure 1.2, where the complete work progress is summarised.

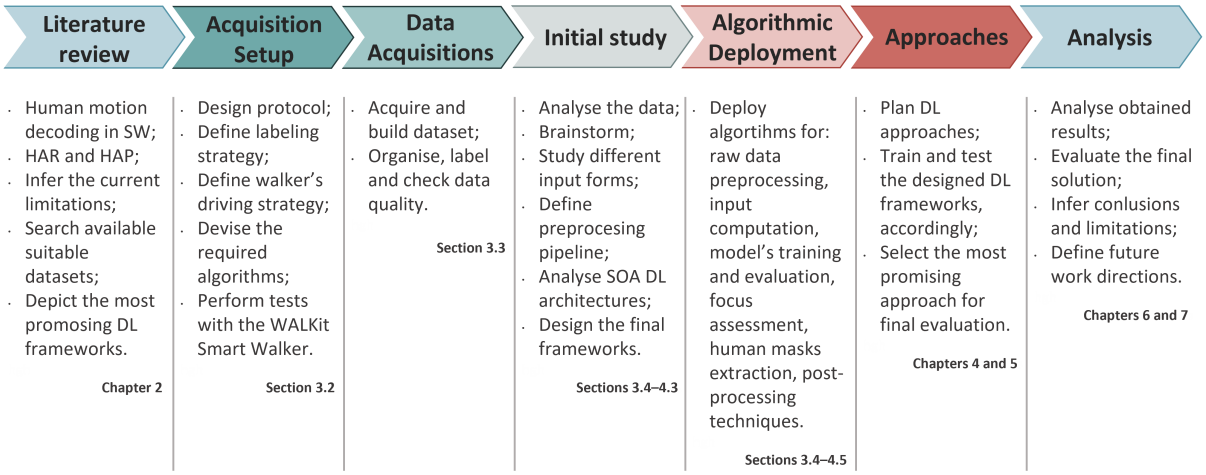


Figure 1.2: Road map of the developed work.

1.6 Contributions

The main contributions of this dissertation to knowledge are:

- Reviews on MI decoding methods currently deployed in smart walkers, as well as on recent DL models used for video analysis, with the purpose of action classification, detection or forecasting.
- A multi-modal dataset suitable for human motion decoding, but also for other tasks outside the scope of this dissertation (for example, gait analysis and human pose estimation).
- Tailored input forms to encode motion information in one single RGB frame, decreasing the required model's complexity.
- Robust devised approaches, from input design to models' architectures, aiming human action prediction and enhancement of the algorithm's focus.

- A method to quantitatively assess the models' human-centred focus, along with an optimised method to compute and correct the required GT masks of the lower human body.
- A DL framework capable of early detecting the different human motions during walking trajectory (stop, walk and turn right/left), only resorting to RGB images from the subject's lower body.

The developed work led to the elaboration of a journal paper entitled "Deep learning-based solution for real-time human motion decoding in a robotic walker" (under revision).

1.7 Dissertation Outline

This dissertation is organised in 7 chapters, as follows.

Chapter 2 presents a survey of MI decoding task on SWs, followed by a deep review on vision-based DL approaches for video analysis, with special attention to the fields of action classification and forecasting. The Chapter finishes with a summary of the findings.

Chapter 3 introduces the target SW and the solution requirements implied by it. It also describes procedures and other equipment for data acquisition, processing and dataset creation, along with the proposed custom inputs.

Chapter 4 clarifies the complete DL frameworks, model architectures and devised approaches. This serves as a guide to better comprehend the obtained results. Implementation details for actually training and evaluation of these approaches, along with the equipment used to train the inherent DL models are also described in this chapter.

Chapter 5 presents the results obtained with the performed approaches. Along the chapter, an approach will be highlighted according to its results and chosen to perform final assessments about its inference time and performance in real-time simulations.

Chapter 6 critically discusses the results obtained, presenting their limitations and possible explanations, along with research suggestions and improvements.

Chapter 7 concludes the dissertation, while providing a brief analysis of the project and its results, along with future research insights.

Chapter 2

Literature Review

The development of a novel HRI strategy targeting the human MI decoding can promote a more personalised and intuitive control over the SW's motion. Nonetheless, such an approach entails several obstacles that need to be overcome. Therefore, it is fundamental to first understand the state of development of these systems and conclude about their limitations and challenges, as well as innovations that can still be performed. Thus, the following chapter presents and analyses the related literature on the subsequent topics:

1. A brief introduction of human MI decoding solutions currently applied in smart walkers and their control systems, describing the sensors and techniques used, with special attention to AI algorithms (Section 2.1);
2. Introduction and clarification of relevant concepts for action classification and forecasting, in the areas of CV and AI (Section 2.2);
3. An overview of the general DL architectures used for human-centred video analysis, from the vision-based input to the deployed model (Section 2.3);
4. Summary of the findings encountered (Section 2.4).

2.1 Human Motion Intention in Smart Walkers

In rehabilitation robotics, HRI is essential to achieve effectiveness in the rehabilitation process, by tackling the challenges of interfacing robot and human in a natural intuitive manner [33]. Detecting the user's MI and using it to improve and adapt the robot's motion is one of these challenges.

An electronic search was conducted on Scopus database, searching for articles that performed human

motion decoding in smart walkers. For that purpose, keywords such as “human motion prediction”, “human motion recognition”, ”motion intention”, ”human-robot interaction”, ”human-robot interface”, ”smart walker”, ”robotic assistant”, “robotic companion”, ”rehabilitation”, ”gait”, ”walk” and “intention” were used. Moreover, since the use of cameras is a key aspect of this dissertation, keywords as “vision”, “cameras”, “video” were also used. Logic operators, as “AND” and “OR” were used to combine the keywords. The search was limited to the articles’ title, abstract, and keywords. A manual search was also conducted considering the references of the selected articles.

2.1.1 Sensors and human-in-the-loop control frameworks

The task of decoding human MI has been approached differently among the existent SW prototypes manily die to the type of used sensors. For this reason, it is possible to group these assistive devices according to the used sensors for this task. Table 2.1 resumes the work presented on the literature about this topic.

Table 2.1: Summary of the hardware and approaches presented in the literature regarding decoding of the user’s motion intentions, during smart walkers’ assistance

Study and Year	Sensory Type	Sensor Position	Sensing Purpose	R/P	Intention to Decode	Approach
[23] (2020)	3D Camera	Walker	Upper body RGB-D image	R	N.M	Camera’s data was used to create a 3D point cloud of the user’s upper body, computing the human position and heading direction relative to the walker, which is then controlled with a PID controller and sway suppression algorithm.
[12] (2020)	LIDAR and IR thermometer	Walker	Legs positions (LIDAR) and orientations (IR thermometer)	R	MF, TR, TL (for each leg)	Tracks the user to provide close-proximity walking safety support and turn according to the user’s intention, through the detection and classification of lower limb gait
[34] (2020)	Multi-channel proximity sensors	Walker	left and right legs’ distance and speed	P	user’s walking intention speed at the next moment	Predict the user’s walking intention speed to obtain the desired movement speed of the robot
[26] (i-Walk)(2020)	RGB-D sensor	Walker	Full body RGB-D image	R	14 different actions (e.g.: stand up, sit down, walk,...)	Recognise activities and gestures to perform human intention recognition, through estimated 3D human pose features
[19] (2019)	Hall Sensors	Walker (Handles)	Handlebar motion manipulation	R	MF, MB, TR, TL	Classify walking intentions through the detection of specific hand movements recorded by the Hall sensors’ signals
[25] (2019)	LRF and RGB-D sensor	Walker	Step length and human’s CoM position and velocity	P	Human trajectory over a time horizon	Unified method for continuous monitoring of each user and adaptation of the robotic platform’s motion accordingly (front-following human-robot coupled motion)
[18] (AGoRA walker) (2018)	Tri-axial load cells	Walker (Handles)	Force and torque applied on the SW	R	N.M.	Computes linear and angular walker’s velocity, through two admittance controllers that use the force and torque applied on the SW
[16] (2018)	LiDAR and IMU	Walker and IMU on the foot	Directional angle and speed of movement	R	MF, TR, TL	LIDAR sensor determines the walking direction by detecting the knees. IMUs are used to obtain the angular rate of gait.
[15] (ISR-Walker) (2017)	Leap motion (2 IR cameras and 3 IR LED)	Walker (below the handles)	Position of several hand points relative to the sensor’s reference	R	MF, TR, TL	A fuzzy logic is used to get the user’s commands, through the sensor’s signals, and control the walker using a PID controller.
[17] (2017)	Pressure sensors	Walker (Handle)	Voltage corresponding to the applied pressure	R	MF and standing	Use the applied pressure as the input of an AdaBoost Classifier to detect user’s intention.
[35] (2015)	Depth camera	Walker	Feet depth image	R	MF, TR, TL	Fast feet position and orientation detection algorithm for smart walker

R / P = Recognition / Prediction; N.A. = Not applicable; N.M. = Not mentioned; MF = Moving forward; MB = Moving Backward; TR = Turning Right; TL = Turning Left

As one can see, a wide range of hardware has been employed to achieve the purpose of MI detection: **i)** force/pressure/load sensors, **ii)** Hall sensors, **iii)** IR sensors/cameras, **iv)** lasers (LRF or LIDAR), **v)** IMU and **vi)** depth and/or RGB cameras.

The traditional way to decode the user's MI corresponds to the use of force sensors. Cheng *et al.* [17] uses three pressure sensors on each robot's handles to measure the force applied by the user, in order to classify two stages (moving forward or stop), while the AGoRA walker [36] uses two tri-axial load cells incorporated in the walker's mechanical structure. From the linear force signal and the torque signal, the AGoRA's admittance controllers compute the linear and angular velocities to be applied to the robot's motors. Other famous SWs, like COOL Aide [37], GUIDO [38], PAAM [39] and UFES [40] also resort to these kinds of sensors to measure the force and/or torque that the user is applying, on the handles. In the case of UFES SW, these sensors are built into the forearms' structures, so it can take advantage of its body-weight support. Despite this, some studies argue about the long-term effectiveness of these force-sensing technologies, since they quickly degrade with time [15]. In fact, to replace this traditional force sensing technique, the ISR-AIWALKER [15] defends the implementation of a totally vision-based approach, using IR cameras and Light Emitting Diodes (LED), which improves the interaction with the user, while also allowing complementary safety measures to be applied.

Other smart assistive devices rely on different sensors, like IR sensors, lasers, IMU or cameras. Weon *et al.* [16] for instance, uses a LIDAR sensor, to detect the lower limbs, combined with IMU, to measure the feet orientation over time. However, this approach presented a large error, compared to Kinect measurements, that still needs to be reduced. Zhao *et al.* [12] combines a LIDAR sensor with an IR temperature sensor instead, to access each foot orientation. With this information, the system is able to learn the user's intention and compute a target position for the walker from it, always ensuring a close distance and parallel orientation between the robot and the human's foot. In contrast to other works, the Neural Network (NN) model used here classifies the collected data, not in the overall walking direction, but in the movement and orientation performed by each foot, at each step. It also incorporates voice control, for when the walker and the patient are at different locations. Despite the success of the latter approach, it demands human-robot close-proximity, preventing the user from choosing and adjusting himself to a comfortable position relative to the walker. Also, computing a different position at each step, instead of the overall walking direction, seems more computational expensive and time consuming.

There are other interesting works, resorting to depth camera [35] for the detection of feet position and orientation, Hall sensors [19] to discriminate between specific hand movements and multi-channel proximity sensors [34] which aim to measure the distance and velocity of each leg and use this to predict the user's walking intention of speed. However, the latter still lacks the ability to perceive the turning feature, a feature that, as shown by Page *et al.* [35], can be extracted from the feet kinematics. Park *et al.*, however, requires specific hand motions to encode each type of walking direction [19]. Even being

intuitive and easy-learned hand movements, it still implies a certain cognitive effort level. As for the use of depth cameras [35], the respective images can be corrupted by the light conditions of many real-world environments.

Recently, some researches have been exploring the use of RGB cameras or sensors for this purpose. As they are already commonly embedded in SWs to perform other functionalities (*e.g.* gait analysis or segmentation), one could seize these to decode the human MI, removing the need for additional sensors. Shen *et al.* [23] uses a RGB-Depth camera to perceive the intended walking direction. An hybrid Proportional, Integral and Derivative (PID) controller with integrated digital practical differentiator is then implemented to calculate the desired wheel rotation angles, being able to track the subject in forward and turning movements, although the use of upper body information entails a disadvantage: the upper body swaying [23]. Chalvatzaki *et al.* [25] complements this type of data (RGB-D) with 2D laser data, in order to perform real-time gait analysis, as well as to track the user's Center-of-Mass (COM) position and velocity. This assembled with the desired human-related robot coupling parameters becomes the input used to forecast the human motion and the evolution of these parameters. Additionally, a novel framework for intelligence assistance, the i-Walk system [26], also points out the potential of using RGB-D cameras, in particular coupled with DL models, to attain an effective activity, gesture and human intentions recognition (Table 2.1).

Despite advances in this field, it is noteworthy that most literature studies validate their algorithms with only a few number of healthy subjects. Therefore, it is still required to extend these studies to a bigger and more diverged population.

2.1.2 Artificial Intelligence-based strategies

Table 2.2 emphasises the intelligent mobility assistance devices which resort to AI techniques to process the captured user's intentions.

Up to the author's knowledge, only 3 works implemented vision-based AI algorithms. Shen *et al.* [23], for example, opts for traditional CV methods. First, it locates the upper body inside a predefined bounding box, then implements an histogram-based filtering scheme to remove noise and extract the human torso, with a region growing method, and finally it computes the human pose parameters, through a quadratic curve fitting. In addition, to address the noise and oscillation from the upper body swaying, an additional orientation signal preprocessing module had to be included, which required the tuning of a suppression width parameter, highly dependent on the subject and turning features.

Table 2.2: Summary of the results achieved, as well as the artificial intelligence protocols used (if applicable), in the literature review of decoding the user's motion intentions, during smart walkers' assistance

Study and Year	AI method	Input	Evaluation	Participants	Metrics	Results/Discussion/ Conclusions
[23] (2020)	N.A.	N.A.	N.A.	1 (healthy)	Tracking error	The human-robot position error yielded a maximum of 3cm and an average of 1cm; robot was able to respond to the step inputs rapidly with no visible overshoot; sway suppression reduced the robot sway by over 50%.
[12] (2020)	k-means algorithm; NN model: two 512-unit hidden layers with ReLU	Legs positions and orientations	N.M.	1 (N.M.)	Error in °	Average orientation error of 5.5°. Walker can change direction according to the user's expected angle. Model's performance not reported.
[34] (2020)	LSTM + FC layer	left and right legs' distance and speed	Train/test split	N.M.	RMSE	The proposed model produced a RMSE of 0.445 cm/s at constant speed and 0.695 cm/s at varying speed.
[26] (i-Walk)(2020)	FC + 2 LSTM layers + FC + soft-max	sequence of human pose features in a temporal window of length T	Train/test split	13 (pathological) and 20 (healthy)	Confusion Matrix	The best accuracy for healthy patients was 95.20%. TurnStanding and Walking activities achieved TP=32.6% and TP=88.5%, respectively. Middle fusion with max-pooling improved the results, as it can detect the most discriminative part of the video clip and ignore the other parts, recognising better highly confusing activity classes.
[19] (2019)	SVM with a RBF kernel	Hall sensors' data	10-fold cross validation	3 (healthy)	Accuracy, Precision, Recall and F1-score	Accuracy and recall surpassed 90% for all classes. Two SVM models: the first achieved 98.9% of accuracy, recall>0.9 and F1-score=0.99; the second one yielded an accuracy of 95.2% and recall of 0.96.
[25] (2019)	2 FC Layer + 2 LSTM	Position and velocity along the x- y-axis and human-related robot coupling parameters (separation distance and bearing)	Train/test split	14 (pathological)	MSE loss	MSE training loss was 4x10-4, while the testing loss was 2x10-3, meaning that the model efficiently forecasts motion intention.
[18] (AGoRA walker) (2018)	N.A.	N.A.	N.A.	1 (healthy)	Graphics	The user's commands and subsequent walker trajectory were in accordance with the ideal path, although higher differences were found at the trajectory corners. The 90-degree turns were more difficult to accomplish.
[16] (2018)	N.A.	N.A.	N.A.	1 (N.M.)	Graphic results	Walker correctly follows the subject in a straight path, but, in the turning movement, the angle of the detected direction is larger than that of the user's movement.
[15] (ISR-Walker) (2017)	N.A.	N.A.	N.A.	5 (healthy)	Graphic results	Walker's motion was in sync with the user's intent, offering no perceptible resistance.The commands were smooth and stable and no significant delay was reported (<200ms). User's found it intuitive and easy to manoeuvre.
[17] (2017)	AdaBoost Classifier	600 labelled input vectors of pressure output	2-fold cross validation	N.M	Accuracy and error rate	AdaBoost Classifier is set to 20 weak classifiers, but 5 was considered ideal. Best model achieved an accuracy of 98%.
[35] (2015)	N.A.	N.A.	N.A.	3 (healthy)	RMSD	Less precision than methods using markers, but it's still faster than using 3D models, robust against clothing variations and continuously detect the feet orientations. Error in orientation is about 20% (approximately, 7°).

N.A. = Not applicable; N.M. = Not mentioned; LSTM = Long short-term memory; FC = Fully Connected; SVM = Support Vector Machine; RBF = Radial Basis Function; MF = Moving forward; MB = Moving Backward; TR = Turning Right; TL = Turning Left; NN = Neural Network; RMSD = Root Mean Square Deviation; RMSE = Root Mean Square Error; TP = True Positives

Both Chalvatzaki *et al.* [25] and the i-Walk platform [26] perform 3D human pose estimation first, using a RGB-D sensor, and then use the pose features as the input of DL models. The former, relies only on the upper body pose, extracting from this the COM motion. Then, two separated 2-layer Long Short-Term Memory (LSTM) models are used to predict the future human COM states, over a time horizon *T*, and the next time step human-robot coupling parameters (separation distance and bearing), respectively. This work innovates by also implementing a Reinforcement learning (RL) strategy, where they use the previous computed information, associated with the estimated stride length provided by gait analysis, to train a policy in charge of proposing control actions to the walker. However, this is a very complex and expensive approach, also because it implies expensive sensory systems, for instance, to train the policy (VICON system). As for the latter, the best results were obtained by exploring the 3D features with body normalisation scheme (BNORM) as input of a 2-layer LSTM network, without Fully Connected (FC) layers. The addition of 3D velocities also boosted the model's performance. Within this approach, a temporal fusion mechanism was also implemented, in particular, the aggregation of the LSTM hidden states with the max-pooling function. Despite the simplest algorithm, the action recognition results still need to be improved. Furthermore, this technique was designed for activity and gesture recognition, not specifically to drive the walker according to the human motions.

Therefore, machine vision inputs and algorithms offer a promising and intuitive way to decode the user's MI, as it is enhanced in Table 2.2. Nevertheless, they still present a lot of challenges. Until now, most of the approaches rely on pose estimation, which implies complex and obtrusive markers setups on the user's body [25][26]. It also requires, normally, two computational tasks, first the pose estimation and then its interpretation or classification as walking intentions, which it is more expensive and prompt to error propagation. As for the algorithms used, the DL methods have grown as an attractive and powerful solution for the MI decoding problem, as they have a superior generalisation capability, without relying on the tuning of subject-dependent parameters, nor on complex filtering techniques [17][19][34].

2.2 Action Recognition and Anticipation

Humans have the capacity of unconsciously predict how other people move around them, by observation. This ability to perceive the environment and recognise patterns helps them anticipate other people's actions or movements and make better decisions based on this interpretation. This functionality would be a refinement of HRI, allowing smart machines to anticipate or at least detect human actions/motions at their early stages and act accordingly [41].

With the progress of AI over the years, there has been substantial research and improvement in the field of HAR [42] or even, more recently, HAP [43] using sensor data. While the former tackles the issue of current action recognition or detection, the latter intends to anticipate the action's ending or even beginning, taking a defiant step towards forecasting. As one can see in Section 2.1, there are several robotic assistant prototypes already attempting to solve similar problems, through the classification of signals from a variety of sensors: force, lasers, cameras, among others. Nevertheless, thanks to its relevant and wide range of applications, HAR and HAP have become a specifically important topic in CV. Recognising, detecting or forecasting actions or motions through videos has an important role in video surveillance, video analysis (in sports, for example), HRI and healthcare [43].

2.2.1 Human Action Recognition

As it can be seen in Figure 2.1, the task of action recognition can be divided into two categories: **i)** *action classification*, which aims to classify segmented videos, each one containing only a single action and **ii)** *action detection*, comprising the spatial and temporal action localisation. Performing this last subtask *offline* includes the detection of start and end times of each action in the video [44], while Online Action Detection (OAD) focus on the problem of localising actions in untrimmed videos as soon as they

happen [32].

Research on the topic of HAR has been resorting to several methods, in terms of data type and algorithms. The most common methods are based on colour (RGB) data [29], on a combination of colour and depth (RGB-D) data [45], or on skeleton data [46]. Until recently, the traditional hand-crafted features with machine learning methods was the basis for action recognition. However, this type of approach is quite vulnerable to camera movement, complex scenes and occlusions, deteriorating the hand-crafted features' quality [44]. For this reason, DL-based methods have become very attractive in the field of CV, as they can automatically learn image features from a variety of data (single-mode or multimodal fusion data), with higher recognition performance than hand-crafted features [44].

Regarding the action detection problematic, this can be addressed offline or online, depending on whether the full video is known before-hand or if it is provided in a real-time manner. The latter is a more challenging and less addressed task. [32]

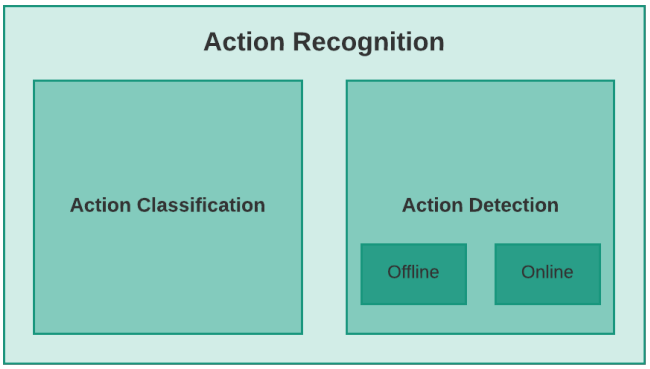


Figure 2.1: Schematic representation of the action recognition field of research.

2.2.2 Human Action Prediction

Action prediction represents a similar problematic to the aforementioned (Section 2.2.1), however a more recently approached and defying one, where action labels are inferred from incomplete observations of the action itself [43]. This task comprises **i)** *action anticipation*, which aims to anticipate the immediate future, without any observation of that future action; **ii)** *early action recognition*, which recognises the action's label from a partial observation of that action and **iii)** *early action detection* that aims to detect an action as early as possible, before its end, from untrimmed videos [43] [47]. The resemblance between the latter and OAD is evident, but early action detection can also include offline performances, assuming various possible observation ratios to classify the action, as long as they comprise its beginning. OAD makes no assumption on the video and must detect the start of an action as soon as it happens. A

schematic of HAP sub-tasks can be seen in Figure 2.2.

The research work developed on this topic mainly uses the same inputs as for HAR (RGB, RGB-D and skeleton data), as these constitute very similar problems. Both tasks could be simplified in two stages: action representation (including, feature representation and extraction) and action classification. To have an efficient model for HAP, this same model has to first perform well at the action recognition task (HAR) [41].

With future prediction arises a new issue: the model's uncertainty. Since the model only gets access to part or none of the observations of the action to predict, one cannot be completely certain about the correct class, as the future comprises multiple possible outputs [48]. For example, following an walking straight, a person as four different possibilities: turn right, turn left, stop or keep walking. Therefore, it becomes necessary, for real-time applications, approaches capable of dealing with this uncertainty.

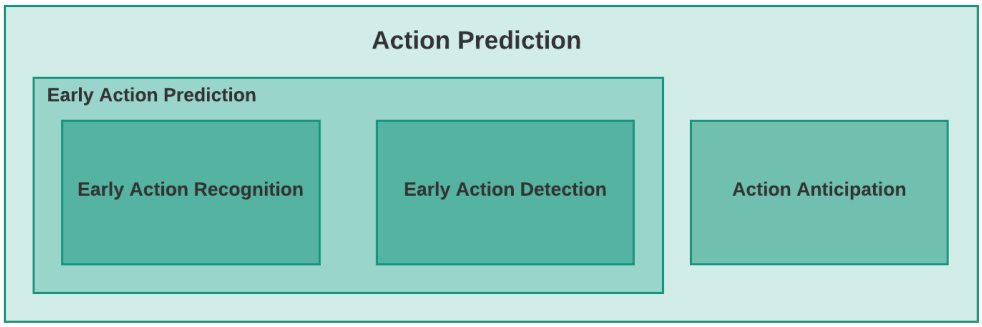


Figure 2.2: Schematic representation of the action prediction field of research.

2.3 Deep Learning methods for Human-centred Video Analysis

Although widely used in many applications, accurate and efficient video analysis, including HAR and HAP, remains a defying area of research in the field of CV, specially online or real-time analysis. Over the years, the use of deep learning methods in this area has increased, as these have the capacity to perform well while automatically learning robust feature representations from raw data (*end-to-end algorithms*).

An electronic search was conducted on Scopus database, searching for articles that performed human action recognition or anticipation, through vision-based DL algorithms. For that purpose, keywords such as "action detection", "action prediction", "action forecasting", "artificial intelligence", "deep learning", "computer vision", "human actions", "fine-grained", "video", "RGB", "classification", "detection", "prediction", "CNN" and "attention" were used. Moreover, as the final aim of this dissertation consisted on motion decong, keywords such as "human motion", "motion intention", "motion decoding", "motion

intention recognition" and "walk" were also used. Logic operators, as "AND" and "OR" were used to combine the keywords. The search was limited to articles' title, abstract, and keywords. A manual search was also conducted considering the references of the selected articles.

2.3.1 Vision-based Inputs

In order to fully explore the whole video content for action recognition or anticipation, different types of input have been used. As mentioned in section 2.2, the most common methods rely on colour (RGB) data, depth data or on skeleton data and combinations between these three.

Depth data appears as an interesting alternative, since it is stable with respect to environment or background changes and also allows the object segmentation, according to depth. However, depth sensors are easily affected by light, outputting large errors and low precision in outdoor non-controlled environments [44]. Laser scanners constitute other option to measure depth, however these sensors are expensive, which would imply an expensive solution.

Skeleton data, on the other hand, has been intensively studied for HAR and has been increasingly attracting more attention. This intrinsic high level representation has been suggested as valuable information for recognising human actions [49]. Also, when compared to RGB-D videos, this type of representation is robust to lighting changes and clustered background [50]. Nevertheless, acquiring skeleton data involves either complex or/and expensive set ups of sensors, such as Xsens, Microsoft Kinect cameras or Vicon, or heavy deep learning models, still in development [51] [52]. In any of these situations, there is always an error associated that will then propagate to the task of action classification. Moreover, this type of input may also not be suitable for recognising fine-grained actions with marginal differences [46].

As for RGB data, these images alone encode static appearance at a specific time instant, but lack the temporal context information provided by the neighbouring frames [53]. Also, they may contain background information not relevant for the targeted action itself. For these reasons, these type of information is normally stacked in windows of RGB frames or combined with other input forms, such as depth frames or skeleton information. In addition, RGB information can be seized to generate complementary forms of input, which describe appearance change and salient the motion between images. One common example is the Optical Flow (OF).

Optical Flow

From the ordinal RGB frames, it is possible to compute other forms of images that can encode/represent the motion between frames, such as dynamic images [54] or OF [55]. These are commonly used

as a complement input of RGB frames, for instance, in two-stream Convolutional Neural Network (CNN)s architectures [55].

The OF exhibits capture motion information and its estimation has suffered impressive developments in recent years, shifting the research paradigm from traditional approaches to deep learning models, namely CNNs [56]. Nevertheless, this field of computer vision still presents many challenges to be overcome, since traditional methods are too computational expensive, making them unsuitable for real-time and mobile applications, while DL techniques require a large amount of data, as well as parameters, which results in huge memory print and may cause overfitting [56].

An alternative suggested to overcome these problems corresponds to the computation of RGB difference [53] between consecutive frames. The resulting image is still capable of describing the appearance change, without the time consumption of traditional OF methods or the DL models' limitations. Despite having an inferior performance when compared to OF, Wang *et al.* [53] presented this alternative as a "low quality, high-speed alternative for motion representations".

2.3.2 CNNs as Feature Extractors

CNNs are widely used in CV, as they are good for natural signals that come in the form of multidimensional arrays, such as RGB-D images. CNNs are now considered highly successful in feature extraction from high dimensional data, constituting the base building block for most architectures in CV tasks, such as image classification.

Simonyan *et al.* [57] proposed a new CNN architecture, namely VGG16, being used in a wide range of tasks and datasets, such as action detection and classification [29]. This model uses very small receptive fields (3x3) throughout the whole network, convolved with the input at every pixel (with stride 1). This allowed the steady increase of the network's depth, through the addition of more convolutional layers. The result was significantly more accurate CNN architectures (at the time), with less parameters, confirming the importance of depth in visual representations.

Inspired by this structure, He *et al.* [58] introduced a deep residual learning framework (ResNets) [58] which learn residual features added then to the input through summation skip connection. The resulting model has less filters and lower complexity than VGG nets, while showing easier optimisation and accuracy gains with greatly increasing the network's depth, allowing the training of deeper and higher complexity models. This boosted in new derived architectures that made use of residual connections, such as Inception-Resnets [59], Densenets [60], among others.

While this architecture evolution enabled superior performance, CNN models remained *black boxes*,

meaning they are incapable of being decomposed into individually intuitive components, making them hard to interpret. Thus, in order to build trust in these intelligent systems and improve them, it is necessary to explain why they predict what they predict, understanding their flaws, failures and successes. Therefore, a class-discriminative localisation technique capable of generating visual explanations for any CNN-based network without requiring architectural changes or re-training was introduced. This technique, named Grad-CAM, produces "visual explanations" for CNN decisions, making these models more transparent and explainable. [31]

2.3.3 Deep Learning approaches

Table 2.3 presents a summary of the state-of-the-art DL approaches for HAR and HAP, including the methods, experimental protocol and most relevant results that allow the comparison between algorithms.

Table 2.3: Summary of the DL approaches, presented in the literature, for HAR and HAP. The studies were organised according to the inverse alphabetic order of the task they are tackling.

Study and Year	Task	Input	Method	Loss	Output	Evaluation	Dataset	Metrics	OR (%)	AT (s)	Results
[29] (2020)	R	Stacks of 7 RGB consecutive frames	VGG16 + Bi-LSTM+softmax	Categorical Crossentropy	Sequence's class (fall, not fall)	5-fold cross validation	Fall Detection	AUC and Confusion Matrix	N.A.	N.A.	Mean Recall of 91.6 and 86.0 for the fall and not fall class, respectively
[46] (2016)	OAD	Framewise 3D streaming skeleton data	3 LSTM layers + 3 non-linear FC layers + Classification (FC+Softmax) and Regression (FC+SoftSelector+FC) Networks	Cross-entropy loss (classification) and Squared Loss (regression)	Frame's class and start and end confidence coeficients	Train/Test split	Online Action Detection (OAD)	F1-Score, SL-score, EL-score; PrecisionRecall Curve	N.A.	N.A.	Maximum average F1-score of 65.5% and, for the forecasting of action's start and end points, low precision and recall (less than 40%)
[61] (2018)	OAD	Framewise RGB data	CNN model (two-stream CNN, VGG-net) + LSTM layer (1st stream)+LSTM w/ feedback loop (2nd stream)+combination unit (+ optional extra LSTM layer)+ FC layer	N.M.	frames's class	N.M.	Breakfast Dataset (BD)	Accuracy over all frames; mAP and cAP	N.A.	N.A.	Accuracy of 32.55%, superior to the use of pose features
[30] (2018)	EAD	RBG sequence	CNN + Feature Mapping LSTM w/ RBF Kernel Mapping + 2-layer MLP appended w/ a RBF kernel layer	Combination of L2 and adversarial loss (regression) and Cross-entropy loss (classification)	Future feature maps and respective class	Dataset's Train/Test split	UCF-101	Accuracy	50	N.A.	Accuracy of 98%
[14] (2017)	EAD	RGB sequence	VGG-16 + context and action-aware submodels + multi-stage LSTM + softmax	Novel designed loss	Sequence's class	Dataset's Train/Test split	UCF-101	Accuracy	1	N.A.	80.50%
[41] (2020)	EAD	Sequence of upper body and object 2D points	2 Deterministic/Stochastic LSTM layers + softmax + decision making criterion	N.M.	Sequence's class	Train/Test split and 10-fold cross-validation	Acticipate dataset	Accuracy	19; 25	N.A.	Accuracy of 95.42% for deterministic LSTM with na OR=19% and of 98.75% for stochastic LSTM with na OR=25%
[62] (2019)	EAD	Sequence of RGB frames	Two-stream CNN (Teacher) + 3 Convolutional layers (student) + average pooling layer, dropout layer and FC layer (Classifier)	Novel weighted loss function	Sequence's class	Dataset's Train/Test split	UCF-101	Accuracy	10	N.A.	92.59%
[63] (2016)	A	Framewise RGB data	CNN (AlexNet) + 2 FC layers + K * (3 FC layers) + SVM	Euclidean loss (regression) and N.M.	K future visual representations + class distribution for each one	Dataset's Train/Test split with 25-fold cross validation	THUMOS	Accuracy	N.A.	1	Highest achieved accuracy = 43.6±4.8%.
[64] (2017)	A	6 consecutive RGB frames	CNN + LSTM-based Encoder-Decoder + 2 FC (classifier) + RL Module	Squared loss (regression) and Cross-entropy loss (classification)	Sequence of future visual representations + sequence's class	Train/Test split	THUMOS	Accuracy, per-frame mAP and cAP	N.A.	1	Accuracy of 50.2%.
[47] (2019)	A	16 RGB frames	I3D CNN network + multi-scale temporal convolutions + attention mechanism	Sum of cross-entropy losses for each recognition and anticipation predictions	Future class	5-fold cross-validation	50Salads	Accuracy	-	1	Accuracy below 70%

R = Recognition; OAD = Online Action Detection; EAD = Early Actio Detection; A = Anticipation; OR = Observation Ratio; AT = Anticipation Time; N.A. = Not applicable;
N.M. = Not mentioned; CNN = Convolutional Neural Network; LSTM = Long short-term memory; FC = Fully Connected; MLP = Multi-Layer Perceptron; SVM = Support Vector Machine;
K = integer number; RBF = Radial Basis Function; mAP = Mean average precision; cAP = Calibrated average precision; SL-score = Start Localisation score; EL-score = End Localisation score

As it can be seen, most of the state-of-the-art approaches exploit the power of CNN as features extractors, as well as the Recurrent Neural Network (RNN)'s ability to model temporal dynamics. Therefore,

the main commonly used network structures for human action recognition, detection or anticipation correspond to two-stream 2D convolutional networks [61][62], 3D convolutional networks [47] and LSTM.

Human Action Recognition

The task of action recognition can be considered as the basis for action forecasting. Thus, to have a good model for action anticipation, this one should first perform well in recognising actions [41].

Berardini *et al.* [29] applied a DL solution to automatically recognise falls in stacks of 7 RGB frames. The model architecture consisted on a CNN model as feature extractor, namely the VGG16, and a Bidirectional LSTM (Bi-LSTM) as feature classifier. As for the training procedure, to overcome the lack of large and annotated publicly available datasets for certain actions (e.g.: turning and falling), transfer learning techniques were implemented, using VGG16 pretrained on the ImageNet dataset and Bi-LSTM pretrained on the UCF-101 action recognition dataset, followed by fine-tunning the Bi-LSTM on a custom-built fall dataset. The approach shows potential, but there's still room for improvements, such as: **i)** exploiting different frame sequence lengths as input and different model architectures, including the use of different NNs and forms of input (like depth images, optical flow, human RGB masks, among others); and **ii)** increasing the size of the fall dataset to help the learning process.

Moving from specially offline action recognition to the OAD task, Li *et al.* [46] tackled this challenge by implementing a LSTM model. Based on skeleton information, a frame-wise action (and background) classification was performed, while simultaneously estimating the start and end frames of the current action, based on the definition Gaussian-like curves for each action. The algorithm consisted on 3 stacked LSTM layers and 3 non-linear FC layers as the feature classifier, feeding its output to two different branches: a classification and a regression network. This achieved a maximum average F1-score of 65.5%, for the OAD task. Performing also the forecasting task of predicting if an action will start or end soon, within an expected time prior to its occurrence, this approach revealed some difficulties associated with future prediction, since actions can have similar poses before they start (*e.g.*, eating and drinking). The authors reported low forecast precision and recall (less than 40%). For these reasons, including appearance features, such as RGB-data, could lead to improvements, as these inputs would give more context information to help differentiating similar positions.

Still concerning the OAD task, Geest *et al.* [61] proposed an approach to model long term dependencies between actions, since human actions sometimes imply a certain order (like standing, after being sited). They used a state-of-the-art CNN model (such as VGG) and then fed its high-dimensional representation into a two-stream LSTM feedback network, where one stream models the CNN's output and the

other models the temporal relations. Here, the use of CNN features is defended over Pose features, as they obtained better performance. Nevertheless, the model overfits for datasets that do not present any dependencies between actions. Thus, it is not suitable when there are no restrictions in the action's order of occurrence. Such is the case for walking trajectories.

Human Action Prediction

Contrarily to the widely studied action recognition problem, the HAP literature focuses on two types of approach: **i)** directly predicting the future frames' class or the current one, before the action ends (classification) [14][41][47][62] or **i))** generating future visual representations that are further classified (regression followed by classification) [30][63][64]. Moreover, a common focus of these studies is to develop novel loss functions that can reduce the predictive generalisation error [14][30][64].

Vondrick *et al.* [63] is one of the authors who resort to visual representations as promising prediction targets, encoding images at a higher semantic level than pixels and without supervision. Hence, unlabelled video is used to learn to predict these future visual representations. The proposed model follows a state-of-the-art CNN architecture (AlexNet), implementing five convolutional layers followed by five FC layers, with ReLu activations throughout the algorithm. To deal with the model's uncertainty, the last three FC present K networks, so each frame results in K future visual representations, one for each plausible future. These representations were then used to train and test standard recognition algorithms, forecasting the future action one second into the future, given only a single frame. The results show the importance of modelling multiple outputs during learning and inference, as well as a gain of 19% over baselines, suggesting the benefits of training deep models to predict future representations with unlabelled videos in the action forecasting task. Still, the higher classification accuracy achieved was 43.6%, much lower than the evaluated forecasting performance of humans, which was around 71.7% for a single subject.

Approaches like Vondrick *et al.* [63] anticipate only a representation of a fixed future time, based on a single past frame's representation, dismissing the history and temporal trend. Gao *et al.* [64] proposed instead a Reinforced Encoder-Decoder (RED) network which takes multiple history representations as input and learns to anticipate a sequence of future ones. The video is segmented into chunks of 6 frames, each of which processed by a CNN model (a state-of-the-art two stream CNN or a VGG16 network) to extract a chunk representation. This is then fed to an LSTM-based Encoder-Decoder, outputting a prediction of the future video chunks' representations, in a supervised manner. The classification of these future representations is handled by two fully connected layers. A distinctive aspect of this work is the use of a reinforcement learning module, whose reward function aims to encourage the system to

make correct predictions as early as possible. RED is jointly optimised by the cross-entropy loss, squared loss and the reward function via a two-stage training process. The Encoder-decoder network allows for sequence anticipation and it outperformed other baselines, such as [63], proving the anticipation power of encoding multiple history representations and anticipating future ones, step-by-step. The strength of CNN extracted features also affects the algorithm's accuracy, that is why using a two-stream CNN (with appearance features and optical flow as inputs) yielded better results than VGG16. For example, when the anticipation time is about 1s, the accuracy achieved for anticipating human interaction action was about 50.2%, surpassing the [63] algorithm by 6.8%.

Such as Gao [64] and Vondrick *et al.* [63], Shi *et al.* [30] also focuses on improving the generalisation capacity of future content, by introducing a novel RNN architecture based on LSTM cells. Aiming to improve the temporal dynamics modulation, the CNN output is segmented into equal size sub-vectors, sharing parameters not only across temporal domain, but also across feature space. Radial Basis Functions (RBF) kernels are used, to capture more complicated dynamics more efficiently. The LSTM cells produce the predictions of each feature element in a future frame, which are then concatenated all back together into a high-level feature vector, classified at the end by a 2-layer Multi Layer Perceptron (MLP) appended with a RBF kernel layer. This model achieved 98% of accuracy, for UCF-101 dataset, although it requires seeing half of the video sequence plus the future representations generated by the RNN to classify each class.

Despite all the research and developments, generating future representations is still defying, as well as time-consuming and prompt for error accumulation [47]. Facing problems like the lack of datasets with accurate labels, being able to train these regression models with a large amount of unlabelled data constitutes an advantage. Nonetheless, generating visual representations is still more computational expensive and, sometimes, the learned representation may not be related to the action itself, as it can be influenced by background or other variables [14]. For these reasons, others have tried to simplify the problem, exploiting different types of features and/or tailored losses to directly predict future classes.

Aliakbarian *et al.* [14] attempts to explore context-aware and action-aware features to attain action anticipation. Additionally, in order to encourage the model to predict the correct class as early as possible, a novel loss is designed to highly penalise False Negatives (FN), while reducing the False Positives (FP) penalisation in the sequence's beginning, prompting sensibility. This approach resorts to the VGG16 model, which connects to two different branches, one to compute context-aware features, encoding global information about the scene, and another to compute action-aware features. These features are then sequentially introduced in a multi-stage LSTM: the first stage classifies only the context input, while the

second one finally classifies the action input merged with the output of the first stage. This yielded an accuracy of 80.5%, while only seeing the first 2 frames of each UCF-101 sequence, and 84%, while seeing the first 50 of UT-Interaction dataset. Despite this algorithm outperformed other state-of-the-art approaches, even when exploiting less frames, there are still a few aspects that must be considered. Regarding the utilised features, the extraction of action-aware features relies on Class Activation Maps (CAMs) [65], which indicate the regions in the input frame that most contributed to the class prediction. Although interesting, this approach leads to some inconsistencies: **i)** first, resorting to these CAMs to enhance the feature maps' relevant regions implied scores obtained by the softmax's weights. Therefore, the CNN models must be trained first and then used to extract the feature maps, in an offline mode, which will then be introduced in the multi-stage LSTM. This is not only inefficient, because it requires classifying each frame before proceeding to the multi-frame classification stage, but also not adequate to online applications; **ii)** secondly, CAMs can be good indicators of CNN focus [65], but that does not guarantee these maps always focus on the action's relevant elements. For example, similar human activities may drive the model to extract features from the surroundings (background movement, due to camera motion, objects, among others), as it is difficult to distinguish the human fine-grained movements. In this approach, CAMs improvement is not taken into consideration during training. Hence, the model's weights are not learned in a way that will enhance action-centred feature extraction and improve the model's focus, but only in a way that promotes accurate classification. Moreover, the novel loss implemented here requires the previous knowledge of the total length of each action, as the time instant that is being classified. Thus, this may constitute an obstacle when tackling early detection in untrimmed videos.

Despite not directly using RGB inputs, Canuto *et al.* [41] presents another interesting work in action anticipation, proposing methods to deal with the model's prediction uncertainty and achieving very high average accuracy (98.75%), using an average of 25% of observations. Here, a small dataset of upper body actions, such as picking/placing or receiving/giving a ball in different directions (left, right, front), is used. These actions can be very similar, especially in their early stages, so body motion may not be enough to correctly anticipate them. As these are movements that imply looking into specific directions and the camera is in a fixed position, gaze or head orientation may encode relevant information to anticipate actions more efficiently and rapidly, as proven in [66] for walking events (walking straight and turning left/right). Therefore, the 2D upper body joint points are extracted from raw RGB frames, including the head joints to provide head direction and gaze information. Object information is also embedded to form the final input, which is fed to a 2-layer LSTM model, followed by a softmax layer. Additionally, two principles are proposed to implement a decision-making criteria for model's predictions: **i)** establish a softmax

probability threshold, which can be combined with a pre-defined minimum number of predictions. So, if the predicted class remains for the next Z observations, with a probability above the threshold, the model can be more confident that is the correct class. The setback relies on setting this hyperparameter, as an inaccurate choice of Z may postpone the anticipation of actions that have no ambiguity problem or even not be enough for those with more ambiguities; **ii)** use and run a stochastic model (*e.g.:* Bayesian LSTM) s times over the same input sequence, computing its uncertainty about the class prediction for the respective observation, which is then compared to an established uncertainty threshold. This new threshold avoided the disadvantage mentioned before and lead to improvements in action anticipation. Nevertheless, both seem reliable inference approaches and perhaps more suitable for untrimmed videos than the one applied in [14], where a frame is classified by leveraging the predictions of all the frames up to its time instant. The latter can thus benefit actions actions which are temporally longer, when classifying online untrimmed videos.

A new line of approaches is also emerging in computer vision and video classification, namely the use of transformers [67] [68] and attention mechanisms [47] [69] [70]. Commonly for fine-grained action recognition, the frame or sequence of frames incorporates irrelevant or redundant information, with no discriminatory property. So, these algorithms guide the model to use *attentional regions*, instead of the whole frame, enhance local features and attain selectively feature fusion. For example, Wu *et al.* [70] implemented channel-wise and spatial attention mechanisms, along with baseline CNNs (VGG16 and ResNet-50) and LSTM. Using dynamic image sequence as input, this approach reached accuracy values over 98%. When comparing to LSTM, transformers can be a lighter and maybe more suitable alternative for real-time [71]. Nonetheless, the study of these algorithms for action recognition in videos is still not fully developed.

2.4 Summary

This section summarises the main conclusions drawn from the literature review presented. Regarding MI decoding deployed on SW, this is mostly implemented in a direct mode, which requires some degree of physical intervention from the user. Such is the case for specially designed handlebars, with force [37]–[40], pressure [17], load [36], IR [15] or hall sensors [19] to decode the user's intents. Few are the studies that aim to implement an indirect mode, in which the walker becomes responsible for analysing the end-user's movement and inferring from this the MI of walking forward, TR, TL, and stopping. In this latter mode, some studies use wearable sensors (e.g. IMU) [16] which may be susceptible to electromagnetic

interference and cause long-term discomfort, while minimising technology acceptance. Others resort to human pose information (position and orientation), requiring complex obtrusive markers/sensors set ups or more computational demands, if one wants to infer them from RGB videos. Additionally, this can also lead to error propagation when feeding the computed poses to a motion decoding algorithm. Therefore, the advances on the fields of CV and DL have not yet been applied to this purpose, resorting to RGB cameras to reduce the need for extra sensors and computation, while enabling a seamless and intuitive HRI.

Additionally, the literature review on human MI decoding, during SW's assistance, also reveals another important conclusion: the encountered studies that resort to an indirect way of decoding human MI focus on lower body information [12][16][34][35]. In fact, since these robotic devices are usually front-following the subject who is grabbing/leaning on its handles, trunk and head positions, as well as orientations, may not provide significant variations to distinguish between walking directions and stop. Also, rehabilitation SW usually requires lower angular velocities, with turns being divided in more smaller steps, which makes it even harder to differentiate between turn left, right or walk straight. Hence, lower body information is the most relevant for this task.

Concerning to HAR and specially HAP from RGB videos, these are defying areas explored in CV and have recently experienced improvements by the implementation of DL algorithms. Nevertheless, it is still an emerging area, where further improvements are necessary to overcome some aspects of realistic environments, such as variety of background, light conditions and objects, subject's occlusions and camera motion. Deploying a system like this in a SW prototype implies dealing, not only with all the aforementioned issues, but also with its computational constraints. Another defying trait is the model's execution in real-time.

With this in mind, selecting DL algorithms should take in consideration the processing constraints available, which greatly reduces the pool of methods that can be applied on the walker, while reaching for the lowest detection errors and the fastest action detection/forecast. Nonetheless, the literature provides important insights on human action classification/anticipation, through RGB-based DL algorithms: **i)** CNN models, optionally combined with LSTM, are widely used and attain good performances; **ii)** actions which imply fine-grained movements, such as human walking motion, have their discriminative information concentrated in certain areas, not in the whole frame. So the search for mechanisms or types of input capable of enhancing the relevant features and/or attenuate the redundant information is essential; and **iii)** anticipating an action before it happens or ends requires modelling the algorithm's ambiguities and uncertainty, through the design of novel losses or the implementation of decision-making criteria.

Chapter 3

Materials and Methods

This next chapter specifies the material and methods used to acquire and process all the data required for this dissertation. This includes **i)** an introduction to the target device in this dissertation, along with the implied requisites (Section 3.1); **ii)** the procedures and experimental protocol for data acquisition, which will be detailed in Section 3.2; and **iii)** the data preparation and preprocessing methods, to create a usable dataset and obtain the samples to train and evaluate DL models, detailed in Sections 3.3 and 3.4.

3.1 WALKit Smart Walker

This project, as well as all its algorithmic solutions, were designed to target the WALKit SW prototype used in rehabilitation [72].

3.1.1 System Overview

Figure 3.1 shows the SW prototype used in this project and its components. WALKit is a four wheeled walker device, with two motorised rear wheels (Figure 3.1D) coupled with an encoder, as well as two passive (caster-wheels) on the front. This allows the SW to move according to the desired direction. This direction is defined by controlling each motor independently, with a specific architecture explained in Section 3.1.2.

The robot can be driven in a passive way or in an active way. The latter integrates three different driving modes: **i)** manual guidance, through an handlebar that aims to directly decode the patient's intention (Figure 3.1I); **ii)** the remote control, where the direction can be defined by the physiotherapist,

for example; and **iii)** the autonomous mode, through environment assessment. The device is fed with two 12 VDC rechargeable batteries.

Aiming to become a mobile gait assessment and evaluation tool, the SW integrates multiple sensors, such as two Orbbec Astra (Orbbec 3D Technology International Inc., USA) RGB+D cameras (Figure 3.1A and C), a laser range finder sensor (Figure 3.1F), ultrasonic sensors (Figure 3.1G) and an external IMU (Figure 3.1K). The cameras record at a rate of 30Hz and provide complementary fields of view (A points to the torso and C to the legs and feet).

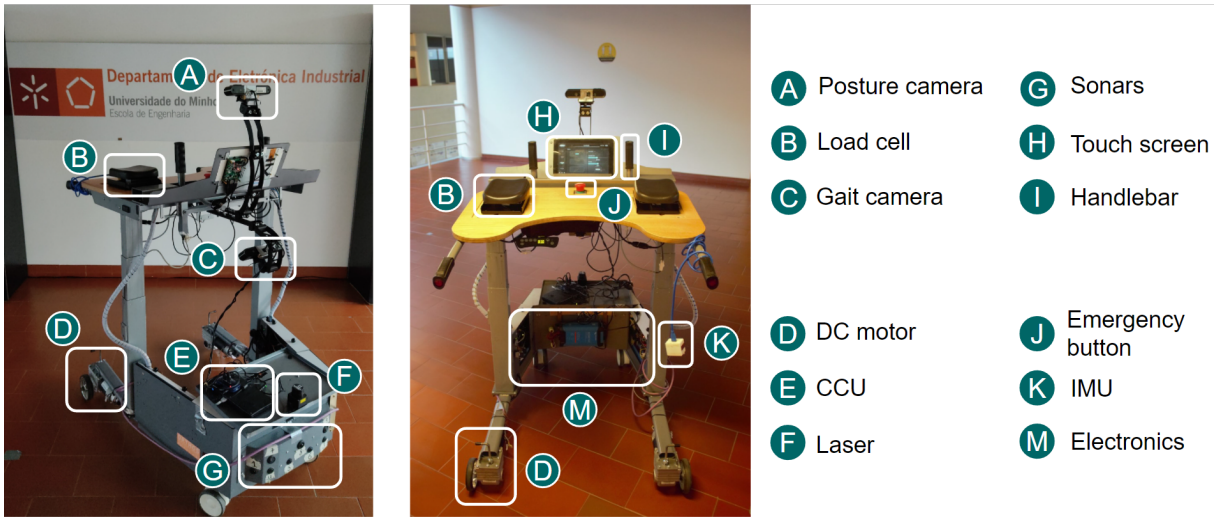


Figure 3.1: Hardware on the WALKit smart walker:

All the data provided by these sensors, as well as the functionalities related to them, can be accessed by both the patient and the clinician, through the use of a dedicated touch screen (Figure 3.1H), that runs a user-friendly graphical user interface (GUI). Additionally, the GUI also allows the therapy setting (by inserting the patient's metadata, activating/deactivating sensors, selecting gait speed and curvature, among others), as well as the activation of other functionalities, for instance, multitasking games or biofeedback strategies.

3.1.2 System Architecture and Functionalities

The software architecture is divided into high- and low-level controls, following a modular and hierarchical architecture. It is of easy interpretation, as well as actualisation, meaning that new functionalities and/or operating modes are easily added.

The low-level control runs a real-time operating system (RTOS) on an STM32F4 Discovery. It is used to operate and read the walker's low-level sensors, such as the load cell (Figure 3.1B), the emergency

button (Figure 3.1J), the IMU embedded on the handlebar (Figure 3.1I), the infrared sensor (not shown in the image), the ultrasonic sensors (Figure 3.1G) and the encoders of each rear wheel (Figure 3.1D). Moreover, this low-level is also responsible for controlling the device's linear and angular speeds through the user's commands, as illustrated in Figure 3.2, by activating/deactivating its motors according a PID controller's response. This controller is expected to adequately produce a response over the computed deviation between the device's measured speed, estimated through the encoders of each wheel, and the reference speed selected on the GUI.

In contrast, the high-level control is responsible for establishing the bridge between humans and the machine (low-level). WALKit SW is equipped with an Intel NUC-6i7KYK (Intel Corporation, USA) mini-computer, running an Ubuntu 18.04 OS with the Robot Operating System (ROS) Melodic Morenia software on top and it is considered the SW's central control unit, since attains the responsibility over all the high-level algorithms and GUI. Only three sensors are connected directly to this level: one laser range finder and two RGB-D cameras. The high-level control communicates with these sensors, under the ROS messaging interface, allowing the assessment of the external environment and the execution of CV-based and DL-based algorithms for gait and posture monitoring.

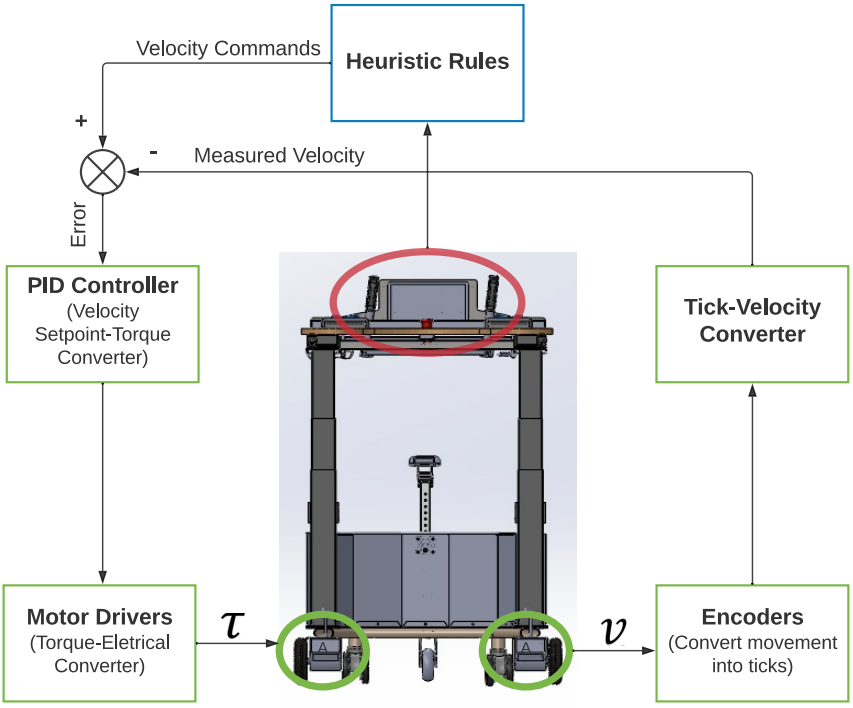


Figure 3.2: Diagram of the human-in-the-loop control strategy currently implemented on the WALKit Smart Walker.

Therefore, the WALKit SW has numerous functionalities, such as: **i)** navigation assistance, with multiple driving modes; **ii)** gait and posture analysis during walker-assisted gait; and **iii)** real-time interaction

applications, such as visual biofeedback and multitasking games. These are important and needed functionalities to follow a human-centred design, where the end-user is mainly involved in the control process, as well as to attain HRI. This helps to deliver a customised therapy, where the patient is encouraged to actively participate in his/her therapy, while his/her residual motor skills are enhanced.

3.1.3 Motion Decoding Requirements

Currently, WALKit SW fosters human motion decoding through an IMU embedded in the handlebar, as show in Figure 3.2. Its outputs are processed and interpreted by heuristic rules to allow the classification and control of the SW's speeds [27]. This approach requires upper-limb coordination and a higher cognitive load, due to the specific hand movements that have to be performed. Besides, it also can present some noise, which can lead to rough motions by the SW.

This robotic assistant also contains two RGB-D cameras, used for gait and posture monitorisation [27], but these functionalities were still not explored towards action/motion recognition or control purposes. Despite the high variability of ataxic gait, these sensors could be a more intuitive and challenging alternative for the patient, since it removes the cognitive and physical burden enforced by the handlebar, while promoting autonomy, as well as more attention to the body's position, orientation and posture. Patients in more advanced stages of rehabilitation therapy could benefit from this approach.

In order to deploy a control strategy through a vision-based human motion decoding solution, several requirements needed to be considered. These were pondered throughout the development of such solution, described in Section 1.5.

During rehabilitation sessions, the SW's high-level computer is tasked with running multiple processes concurrently, which makes it responsible at all times for their correct functioning and patient safety. Moreover, it does not contain hardware accelerators (e.g GPU), normally used to speed up NNs execution. Therefore, these are relevant limitations to be taken into consideration when developing algorithmic solutions for the WALKit SW, as they impose constraints on memory and solution's complexity.

Furthermore, the inference time of the proposed solution, from the input computation and preprocessing until the post-processed output class obtained from the best model, should be ideally inferior than 0.067s (time to record one frame at 15Hz). In this way, the walker's controller would be able to actuate on each prediction in time, before moving forward to the next one.

3.2 Data Acquisition

Several public datasets for action recognition and/or anticipation were considered, like UCF101 [73], MAD [74] and UWA3D Multiview [75]. Nevertheless, these never presented all the features needed to accomplish the aim of this dissertation, since most of them do not include walking or turnings as different actions, nor mobile cameras that follow the subject and always maintain a front-view perspective. Therefore, a custom dataset was created. Data acquisition was conducted under the ethical procedures of the Ethics Committee in Life and Health Sciences (CEICVS 147/2021), following the Helsinki Declaration and the Oviedo Convention. All participants gave their informed consent to be part of the study. Data were collected at the School of Engineering of University of Minho.

3.2.1 Participants

Fifteen healthy participants (nine males and four females) were recruited and accepted to participate in this data collection. A list of inclusion criteria was outlined to conduct the experimental data collection. Participants were recruited if they had: **i)** 18 or more years old; **ii)** body mass within 45 and 90 kg; **iii)** height within 150 and 185 cm; and **iv)** healthy locomotion. Table 3.1 presents the participants' detailed anthropometric data.

Table 3.1: Metadata of the participants included in the acquired dataset with the WALKit Smart Walker

Participant ID	Gender (M/F)	Age (years)	Body height (cm)	Body mass (kg)
01	M	24	170	74
02	M	31	174	68
03	F	22	159	56
04	F	29	157	53
05	M	24	170	78
06	F	24	159	48,2
07	M	26	181	71
08	M	26	175	61
09	M	26	175	61
10	F	24	170	62
11	M	27	175	83
12	F	28	159	68
13	M	23	174	64
14	M	22	169	72
15	F	23	160	58
Mean and STD	-	25.27 (± 2.54)	168.47 (± 7.42)	65.15 (± 9.22)

3.2.2 Mobile Acquisition Setup

Data acquisition was performed outside a controlled laboratory space, using a mobile setup [76]. This allowed the recording of data in realistic scenarios, with non-ideal light conditions and dynamic backgrounds.

This setup is composed by: **i)** WALKit prototype, responsible for recording the visual information from its two embedded RGB+D cameras, at a frame rate of 30 Hz, and for sending start/stop triggers to external recordings; and **ii)** the Xsens MTw Awinda (Xsens Technologies B.V., The Netherlands), which recorded data at 60 Hz. This latter device included a base station connected to a laptop running the MVN software from Xsens.

3.2.3 Instrumentation

Figure 3.3 illustrates a random participant with the Xsens MTw Awinda sensors and the respective mobile setup.

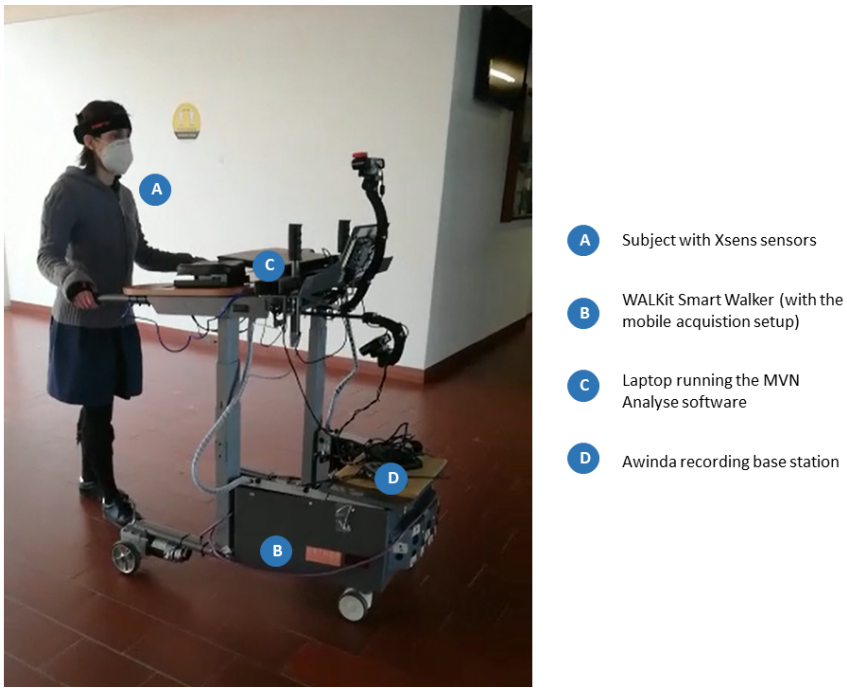


Figure 3.3: Mobile acquisition setup, where a laptop running the Xsens MVN software and its acquisition base are placed over the smart walker, moving along with the robotic device. The user is equipped with the Xsens sensors, hidden bellow a layer of clothing.

The participants were instructed to comfortable clothes and standard shoes to accommodate the on body sensors. Each participant was instrumented with seventeen IMUs from Xsens MTw Awinda.

They were placed on the head, shoulders, chest, arms, forearms, wrist, waist, thighs, shanks, and feet, according to the manufacturer's guidelines[1]. The participants were, then, instructed to wear long clothes covering the sensors to not produce a bias effect while training the models.

3.2.4 Acquisition Protocol

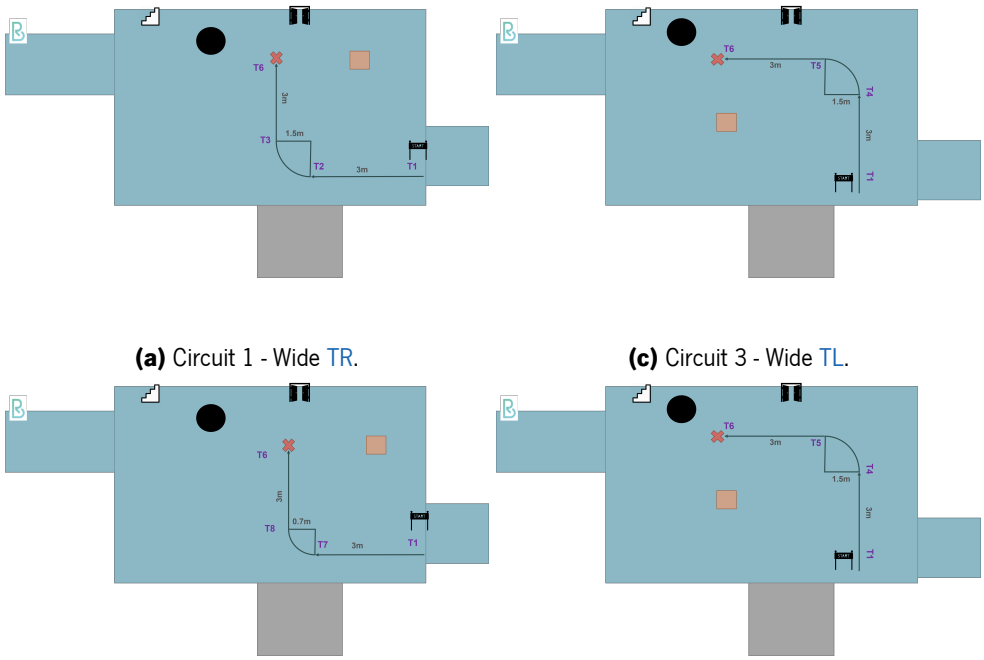
Considering the aim of decoding the human MI while walking straight, turning right, turning left and stopping, it would be ideal to record these natural movements, during a non-predefined circuit. Nevertheless, the final application consists on controlling a SW device, which has a specific turning strategy and restrains the user's gait. So, it would be preferable to use this walker in a passive mode. However, considering the size and weight of the device, it was considered that users would not walk normally when driving the device themselves, causing abnormalities in the participant's gait. For these reasons, an automatic driving mode capable of following given trajectories was necessary, so the SW could be used only as a recording device, that follows the user. This driving mode enables to create velocity commands for each wheel according to the desired linear and angular velocity of the robot and considering the turning radius. This is further explained in 3.2.6.

The participants were instructed to walk with WALKit SW performing 4 circuits, according to the direction (right or left) and curvature's degree (wide and tight). As illustrated in Figure 3.4, the following sequence was performed for each circuit:

1. 10s standing;
2. 3-meter walking;
3. one turn equivalent to ¼ of circumference, with its direction and radius implied by the circuit: right/left, wide (R=1.5m)/tight(R=0.7m);
4. 3-meter walking;
5. 10s standing.

The floor was marked with tape and signalised, during the records, with chairs and staff people, so the participants could see and react to the circuit's morphology. Different light conditions were held for each trial. This helps increasing the environment and movement variability (backgrounds, lighting, velocities, turn features, etc), while improving the statistical significance of the data. In total, 8 sequences were drawn (2 different trials x 4 circuits).

[1] https://bit.ly/31iUaA8



(a) Circuit 1 - Wide TR.

(c) Circuit 3 - Wide TL.

(b) Circuit 2 - Tight TR.

(d) Circuit 4 - Tight TL.

Figure 3.4: Designed 4 circuits, distinguished by the turn direction and curvature radius.

The sequence's content remained unknown to the participants until the beginning of the trial, when a brief explanation of the circuit was given. Also, the respective marked spots were positioned in a way that remained invisible to the SW's cameras, during the performance of each respective trial. This was valid for every sequence, avoiding a biased training where the input data would contain marks for each turning point. Additionally, the overall circuit and trial's order was randomised for each subject before the beginning of the acquisition.

3.2.5 Data acquisition

Each participant performed 2 valid trials per circuit, considering three gait speeds: 0.5 m/s, 0.7 m/s, and 1 m/s. These speeds were selected according to the walker's most commonly used velocity range, as well as considering the typical self-selected slow, normal and fast walking speeds for healthy subjects [77]. In the end, each subject performed 24 trials, taking no more than 1 minute per trial.

Each trial was conducted as follows: **i)** the walker was placed on the starting position of the respective trial; **ii)** the user was instructed to stand in front of the robotic device, in the *N-Pose* position, to reset the Xsens internal referentials; **iii)** the subject grabbed both of the SW's handles; **iv)** a remote controller started the rehabilitation session, as well as the cameras' recording and the Xsens MVN software, through an hardware trigger; and **v)** the participants started the trial acquisition. At the end of the trial, the SW

was positioned at the beginning of the next trial, repeating the process.

The participants were instructed not to just follow the robotic walker, but to interact with it, along the designed circuit, in order to capture their intention as naturally as possible. Some familiarisation trials were performed before the real recording procedure, encouraging the HRI.

3.2.6 Automatic trajectory mode

The *automatic trajectory mode* was implemented in *C++* language, within ROS architecture. Since WALKit SW is a differential drive robot, its movement is controlled by providing independent velocity to each wheel. Considering the differential drive kinematics [78], the velocity of each wheel is calculated as:

$$\begin{bmatrix} V_{left} \\ V_{right} \end{bmatrix} = \begin{bmatrix} 1 & -(l/2) \\ 1 & (l/2) \end{bmatrix} \times \begin{bmatrix} v \\ w \end{bmatrix} \tag{3.1}$$

In equation (3.1), w and v represent the robot's angular and linear velocities, respectively, l represents the total distance between the two wheels and V_{left}, V_{right} the linear velocities each wheel needs to take to accomplish the stipulated values of w and v and, therefore, control the walker's trajectory.

Several experiments to define the most natural turning strategy in an human perspective were considered. A rotation around an instantaneous centre of curvature (ICC), as illustrated in Figure 3.5, was then selected for the SW to perform. In this way, the curves are more intuitive for humans, plus it is possible to define different radius of curvature, allowing to build trajectories with wider and/or tighter curves.

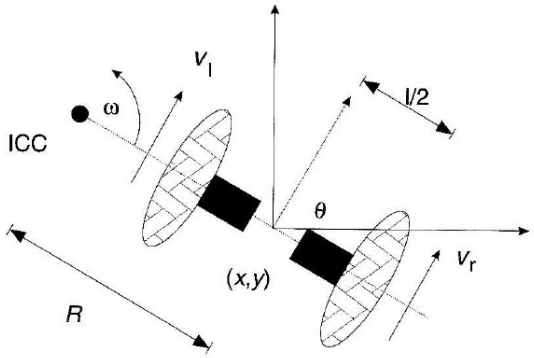


Figure 3.5: Differential drive kinematics. Retrieved from [78].

Since $v = wR$, the system of equations in (3.1) can be rewritten as follows:

$$\begin{bmatrix} V_{left} \\ V_{right} \end{bmatrix} = \begin{bmatrix} w(R - (l/2)) \\ w(R + (l/2)) \end{bmatrix} \tag{3.2}$$

Through this algorithm, turn area, direction, angle (90°) and strategy were controlled, to fulfil the SW's control purpose and also due to the space limitations encountered during the circuits designing and performance.

3.2.7 Labelling

The labelling process was executed in real-time, along the data acquisition, in two different ways: **i)** with joystick commands and **ii)** with velocity commands. The former relies on an external person who uses the joystick's digital buttons, similar to [79], to mark transitional moments between actions, according to his/her observation of clear feet movements. When the variations in the subject's gait were not clear for the naked eye, the transitional moments were marked at most when the participant reached the local of transition defined previously in the circuit's trajectory. This method produced labels responsible for denoting the subject's interaction and intention (*JOY labels*). The latter represents the device's actions, generating labels mainly for action recognition, when the background is already changing accordingly to the performed movements (*VEL labels*).

Nevertheless, both of these labels present some disadvantages. The *JOY labels* are biased by the third person's perception, while the *VEL labels* are always a bit earlier than the SW's actual movement, since they correspond to the PID's reference and not to the actual wheel's velocities. Also, the walker's accelerations/decelerations, as well as some delays inherent to the hardware, may cause some small inconsistencies between these labels and the actual robotic movement.

Facing these disadvantages, foot contacts obtained with Xsens MTw Awinda were used to better position these labels. This helped to determine gait events, such as Heel-Strike (HS) and Toe-off (TO), which were used to identify the beginning of a walk, stop or turn class. The first walk and the last stop events were marked by the first hell-off and last HS of the trial, respectively. As for the turn event, literature considers that the direction of a step is determined and becomes unchangeable at the TO, so including data from this moment on increases the changes of correctly predicting this direction [13]. Thus, to assure the inclusion of this gait event, this type of labels (*Xsens labels*) delimit turns on the HS moments immediately preceding the corresponding *JOY labels*. Although promisingly more accurate, these labels presented failures, due to the Xsens system instabilities, that caused significant perturbations on several trials' recorded foot contacts.

3.3 Dataset

3.3.1 Data Preparation

The generated labels and data obtained from the Xsens are temporally synchronised with the camera streams. This procedure is executed offline, through the timestamps saved during data recording for each one of the modalities and for the hardware trigger sent by the walker to the Xsens base, marking the data acquisition's start. Moreover, this process sub-samples the Xsens skeleton data to 30Hz to match the samples from the camera streams.

All the data was manually and briefly inspected, checking the quality of the visual information and the temporal correlation between these and the respective labels.

As expected, the depth images were corrupted by high infrared exposures from sunlight and only subtle changes were revealed by the user's torso, when walking with the WALKit SW. Therefore, the selected raw data to create the final inputs corresponds to the RGB image from the SW's lower camera, capturing only the legs and feet.

3.3.2 Dataset of Frames

Considering this project's ambition of directly decoding human motion from visual information, the datasets to train and evaluate the different DL frameworks were built with only lower body RGB labelled frames.

Due to space limitations, the duration of turn events in the performed circuits (Section 3.2.4) is always inferior to the other events, leading to an unbalanced dataset. To tackle this problem, a balanced dataset of frames was created, where, for each trial, a sequence of frames per class is extracted. The number of samples was limited to the lower number of frames present in the turn events. To cover more time of action with a lower number of similar samples, given the little additional motion information added by those, the dataset was down-sampled to 15 Hz. This was performed since it was found that the principal walking frequency is no higher than 2 Hz for gait speeds above 1 m/s [80].

Therefore, sequences of 40 consecutive frames were extracted after the down-sampling. Moreover, due to the presence of bias during the labelling procedure (Section 3.2.7), this extraction was performed avoiding the action's boundaries, in order to prevent the risk of catching frames from the previous/following class. For this reason, the start of each class was marked with the latest of the two labels (*JOY* and *VEL*) and a threshold of frames at these boundaries was used, when possible. Considering these constrains,

a balanced dataset containing a total of 28800 RGB-D frames from the SW's lower camera was created, without including transitional frames. Note that the depth images are only included for human masks computation purposes, which will be described in Section 3.4.3.

To further simulate real-time scenarios, a test dataset was created including data of 3 participants with non-corrupted data. This dataset contains the complete trials of frames and is not separated into 40-frame class sequences. Thus, it has 72 untrimmed RGB video trials, labelled with the *Xsens labels* to more accurately mark the transitions.

3.4 Data preprocessing

After data preparation, followed by the creation of a balanced dataset of labelled RGB frames, the images are preprocessed, in order to form the final input that will be fed into the model. This process is documented below.

3.4.1 Input Frames

As the proposed task is to distinguish actions in videos focusing the subjects' legs and feet, the main feature to attain this goal would be, not the presence of objects, their shapes or the pixels' intensities, but the motion and the different body orientations along the recorded frames. Thus, it seems only natural to focus on these particular aspects, so the model can accurately classify distinct walking actions, while focusing on the right and most relevant features.

Literature reveals the use of windows of images as input [29] or other forms of images, usually as complement to the original RGB frames, as it is the case of the OF [55]. Considering this, two different types of inputs were proposed: **i)** the difference between the last RGB image and the first one, considering a sliding window approach (*DIF* input); and **ii)** the sum of all the RGB images in a chosen window, also considering a sliding window approach (*ADD* input). Through experimentation and visualisation, the window length was defined as 4, considering the down-sampled dataset, since it incorporates significant motion across all gait speeds, while never containing information from more than 1 different step. These kinds of input attain a single-frame classification approach and the respective computation process is illustrated in Figure 3.6.

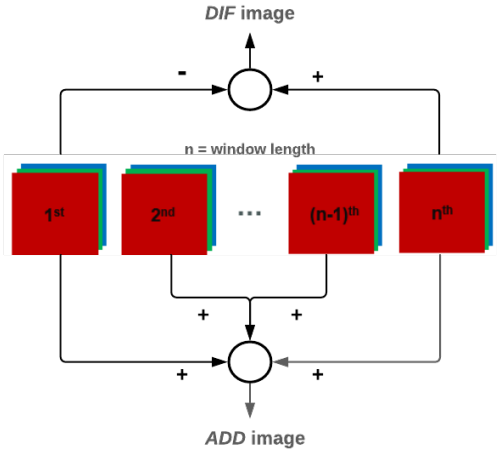


Figure 3.6: Schematic representation of the input computation.

The sum of frames appears here as an original idea to represent the motion along the video, without the computational expense of using a window of frames and heavier models (such as 3D-CNN or a RNN). The difference between frames was already proposed by Wang *et al.* [53] as a lighter and faster alternative to the OF. In fact, running experiments on a Google Colab instance (see section 4.5.2) with the created dataset confirmed that OF computation, using the PWC-Net model from [81], requires 275x more time than subtracting two images and almost 17x the time of the whole pipeline, described in Figure 3.9. Comparing with the *ADD* input, this difference is about 60x and almost 11x, for its computation and the whole pipeline respectively.[2]

Examples of these kinds of inputs are shown in Figures 3.7 and 3.8, where the latter helps to better understand the correlation between the original frames and the resultant *DIF*/*ADD*. For visualisation purposes, the input images were normalised between 0 and 255, after their computation (Figure 3.6). As it can be seen, the subject centralisation and front-view perspective, typical of these SW's recorded videos, allowed the generation of these inputs, where it is visible the change in position and orientation of both feet and legs, from the third past frame to the present one. As for the *DIF* images, the two different body positions that correspond to two different time steps (the present and the first past frame, in the depicted window) stand out, while, in the *ADD* images, the temporal drag of the human position is noticeable in a single frame. Due to camera motion and the pavement characteristics, background movement is perceived too. This is true except for the STOP class, where, as expected, no substantial movement is recorded. However, there are still some light variations and residual human motions which may provoke some low intensity noise, more visible when in a scaled *DIF* image.

[2]The pipeline here does not include the augmentation and normalisation steps, as these are performed on the fly

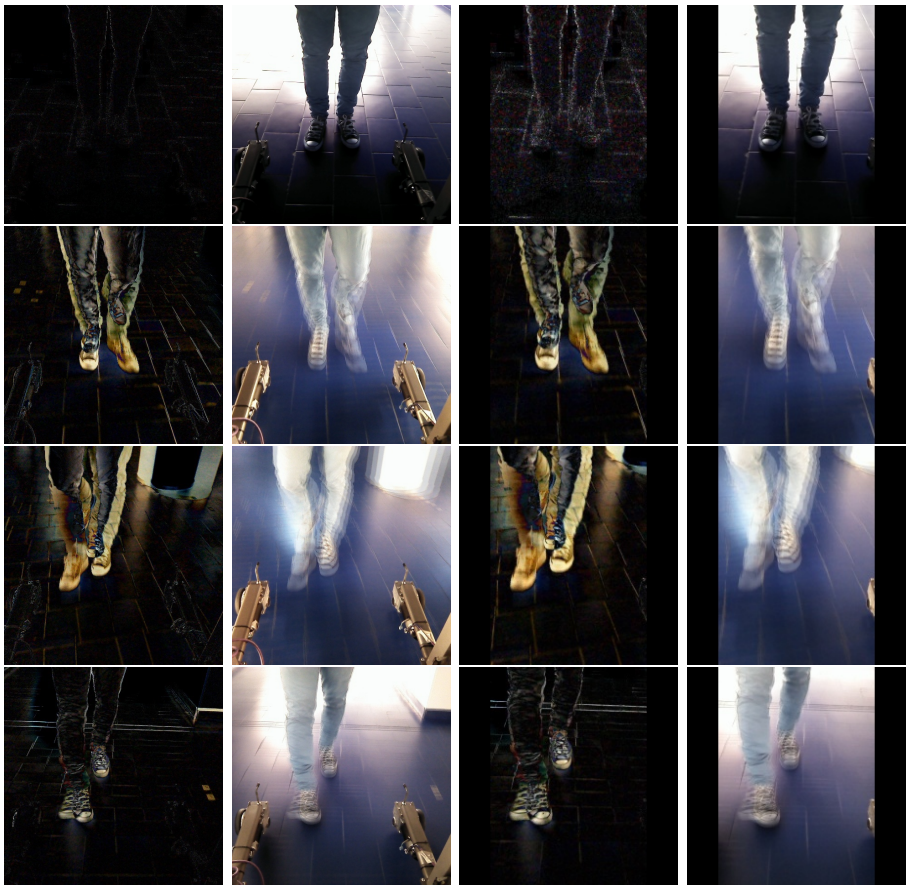


Figure 3.7: Examples of computed **a)** *DIF*, **b)** *ADD*, **c)** cropped *DIF* and **d)** cropped *ADD* images for each class (STOP, WALK, TR, TL, from top to bottom), from the same subject (window length=4). Each row contains images from the same class, trial and time instant.

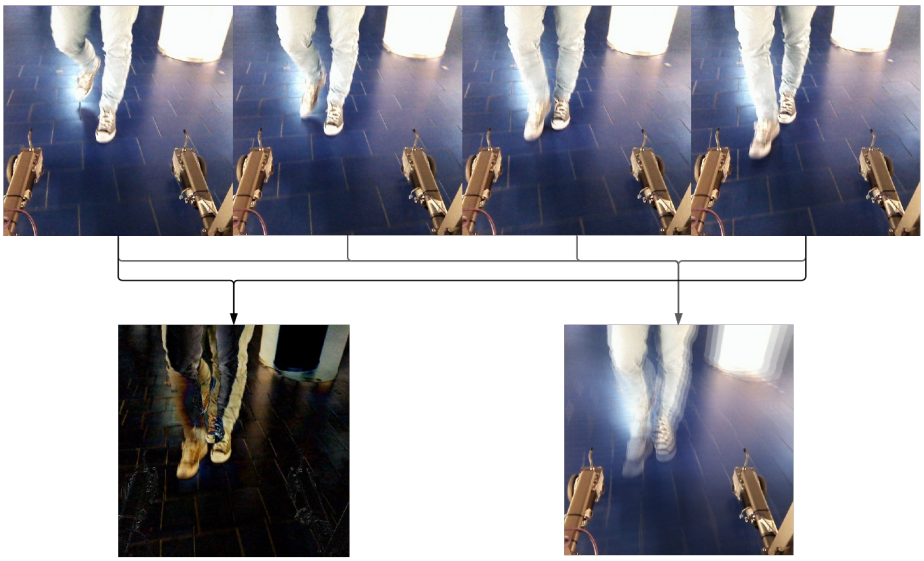


Figure 3.8: A turn right sequence extracted from the created dataset of frames, temporally ordered from left to right and followed by the computed *DIF* and *ADD* images, for a window length of 4.

3.4.2 Preprocessing

The preprocessing pipeline is shown in Figure 3.9 and includes resizing, data augmentation and normalisation. These preprocessing techniques are common procedures shared by every type of input used. Before that, the computed *DIF* and *ADD* inputs may pass by a process of cropping.

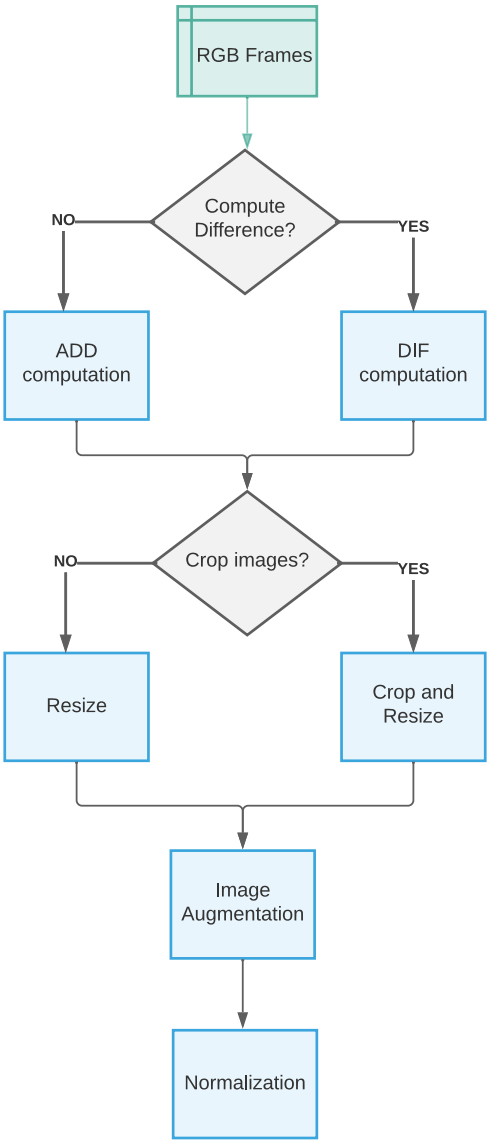


Figure 3.9: Schematic representation of the preprocessing pipeline.

Cropping

The cropping procedure is optional and intends to diminish the input's background area, in an attempt for the model to direct its focus to the user and his/hers fine-grained movements, extracting more meaningfully features. The images are thus manually cropped, according to a predefined Region of Interest (ROI), easily designed, since the user is normally in a central position, in the middle of the walker's

handles.

Resizing

Frames are resized from a resolution of 480x480 pixels, to a resolution of 224x224, preserving the images' aspect ratio. This also happens for images that have been cropped, although the crop procedure decreases the original values of resolution and aspect ratio. So, in this case, the resizing process still preserves the aspect ratio, as best as possible, but there are extra pixels that are set to 0, while the image is centred, as one can see in the last two columns of Figure 3.7.

This resolution of 224x224 was chosen to match the input dimensions of CNN models pre-trained on the ImageNet dataset (see Section 4.5.2). Additionally, this reduction leads to a decrease in computations and, consequently, in inference time, increasing, at the same time, the percentage of the image covered in the model's Effective Receptive Field (ERF), without requiring a deeper model. The loss of fine details should not be significant to the point of compromising the performance, since the participant is expected to stand close to the camera, at all times, while using the SW.

Augmentation

Image augmentation is here used in order to avoid overfitting, as well as improve the model's focus on the subject's body, despite its global position in the images. Spatial augmentation is also very important, when the images are cropped. Changing the position of the back surrounding pixels reduces the chances of these pixels affecting the learning procedure and, subsequently, the model's performance.

This procedure occurs *on the fly* through the use of *ImageDataGenerator* class from *Keras*. Random alterations were applied to the image brightness and contrast, as well as spatial augmentations, such as height and width shifts and zoom. Since directions are an important feature to distinguish between turn right, left and straight walking, rotations were avoided. Moreover, random Gaussian blur was also added.

Normalisation

A stable training of NN requires normalisation of the input. Here, normalisation was preferred over standardisation, preserving the data distribution. Additionally, *DIF* images can have very low intensities and thus dividing them by the standard deviation could lead to excessively high values, causing a great discrepancy between pixels' intensities that could difficult the training procedure. The images where thus normalised between 0 and 1. Examples of augmented and normalised input images can be visualised in Figure 3.10.

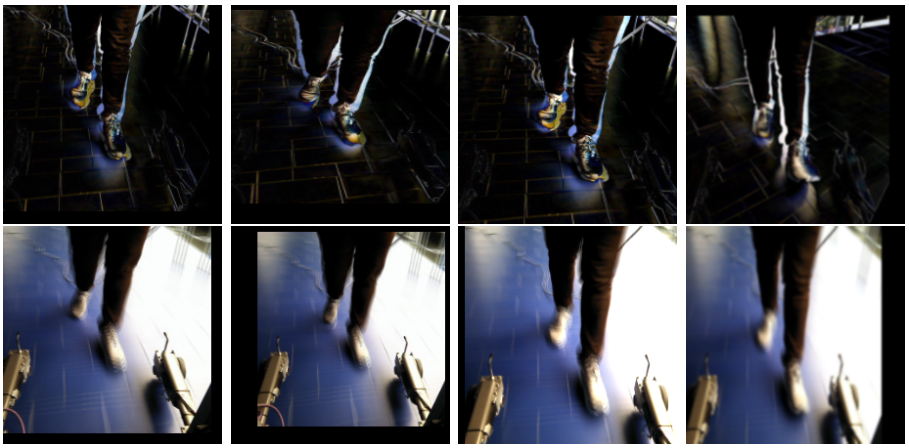


Figure 3.10: Examples of augmented and normalised *DIF* and *ADD* inputs: **a)** with height and width shifts, **b)** with added zoom, **c)** with brightness variations too and, finally, **d)** with contrast variations and Gaussian noise.

3.4.3 Human Masks

To detect the walking directions (straight, left, right) and standing/stopping events, before the user transits to another action, it is necessary to distinguish human fine-grained movements. Therefore, the human body should be the main focus of the deployed model, extracting, for example, relevant features from feet position and orientation.

Therefore, human masks were computed through a classic vision algorithm, illustrated in Figure 3.11. This algorithm involves geometric and threshold operations that remove the background, as well as the floor plane, to isolate the user. The floor plane is computed from a background image, captured by the walker's lower camera, and a floor depth tolerance (FDT) is initially set to 0.05. This value is used to eliminate noise, such as the walker's legs or some floor portions that do not possess the same depth levels. For this reason, an adaptive threshold is implemented at the end, over the whole computed mask: if the percentage of foreground is higher than 15% of the whole image, the mask will be re-computed with a higher FDT (the higher this value, higher the number of pixels recognised as floor, reducing the unwanted noise).

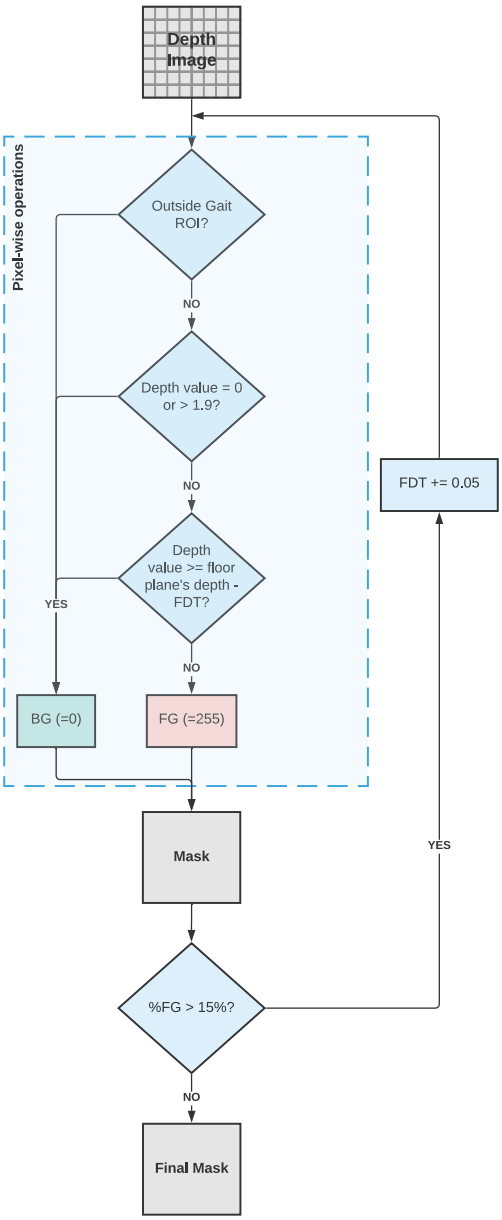


Figure 3.11: Algorithm's flowchart for mask computation.

Using the aforementioned algorithm, all the masks are first computed from the dataset's depth images. Considering that these depth frames could be corrupted by the high exposure from sunlight, another foreground threshold was included to previously discard RGB frames and the respective corrupted masks from the dataset. This threshold is calculated relatively to a designed ROI, both pre-defined by empirical experiments. After these procedures, a final dataset of masks is created and used in a process similar to the RGB input computation (Figure 3.6). A sliding window of length 4 is applied and the relevant masks are selected: the first and last one for *DIF* and all the *N* frames in the window for *ADD* generation. Despite the implemented thresholds, these masks can still present some minor corruptions (*e.g.:* incomplete feet) or even some extra noise (*e.g.:* SW's wheels), which need to be corrected. Thus, a classic vision

algorithm was designed (Figure 3.12), where **i)** the ROI is firstly extracted from the original mask, in order to remove possible walker's legs or side noise; **ii)** then, an opening operation is performed to remove points of noise; **iii)** following this, a closing operation is used to restore mask boundaries; **iv)** a binary hole filling is executed to close possible holes on the human's feet; and, finally, **v)** a second opening process is performed to reduce subsequent dilated boundaries.

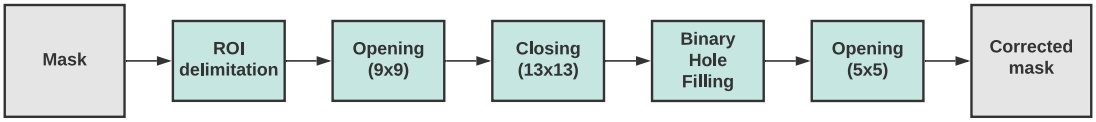


Figure 3.12: Algorithm's flowchart for mask correction.

Finally, the selected and corrected masks are summed and then cropped, according to the input's preprocessing (Figure 3.9), in order to form the final corresponding masks. The complete process of masks extraction is presented in Figure 3.13.

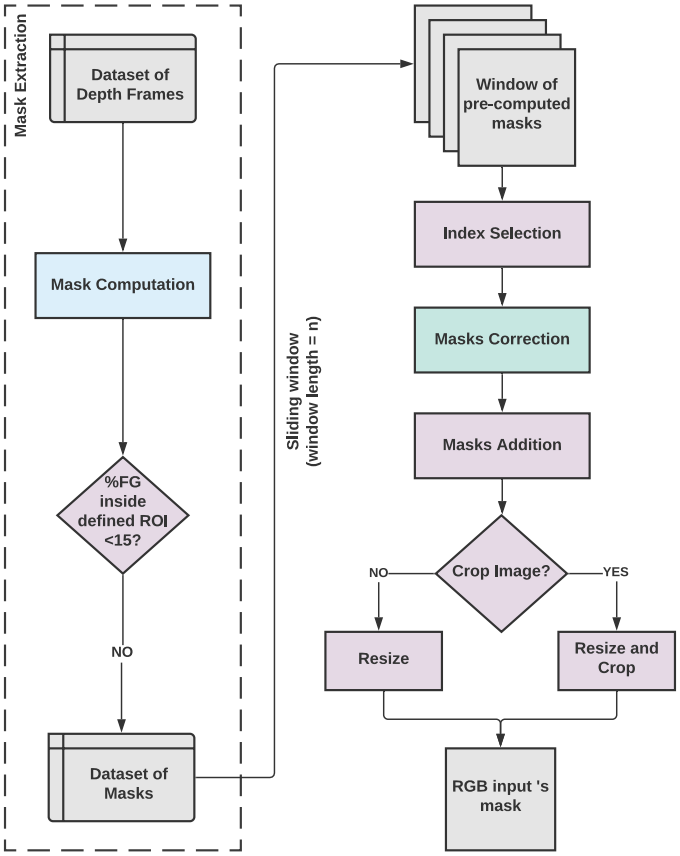
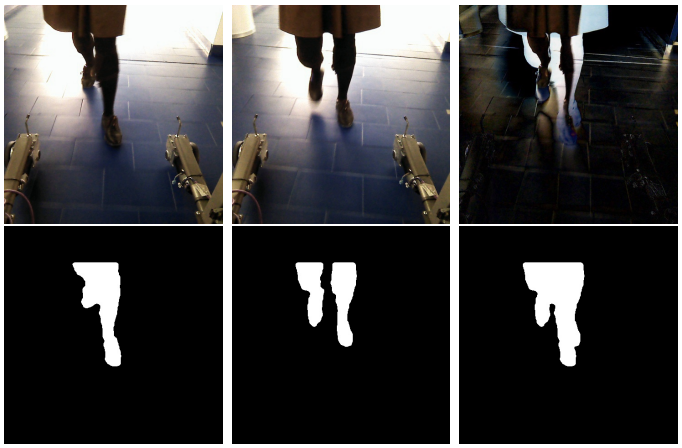


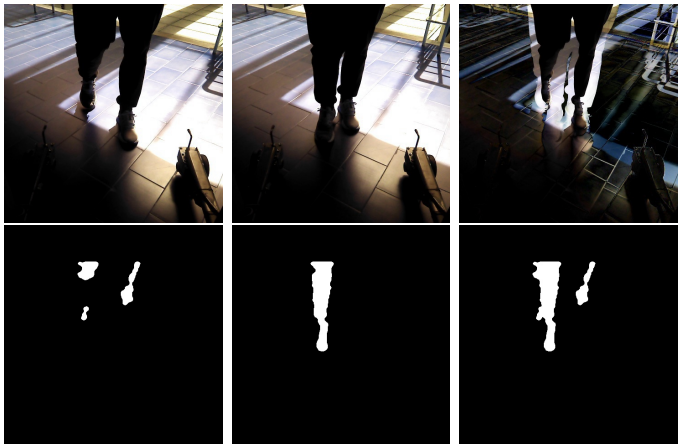
Figure 3.13: Pipeline of the procedure for mask extraction and respective dataset creation.

Examples of the obtained masks are given in Figures 3.14 and 3.15.



(a) Frame -3. **(b)** Frame 0. **(c)** *DIF*.

Figure 3.14: Examples of individually non-corrupted masks (second row), along with their corresponding RGB inputs (first row), for a window length of 4 frames. The presented masks are already processed, as depicted above.



(a) Frame -3. **(b)** Frame 0. **(c)** *DIF*.

Figure 3.15: Examples of individually corrupted masks (second row), along with their corresponding RGB inputs (first row), for a window length of 4 frames. The presented masks are already processed, as depicted above.

Chapter 4

Deep Learning Frameworks

This chapter describes the five DL architectures proposed in this dissertation, along with a description of the three devised approaches (Section 4.1) and the two complete frameworks used for this purpose (Sections 4.2 and 4.3). It also describes the post-processing implemented for online evaluation (Section 4.4), along with the details to train and validate these approaches (Section 4.5).

Considering the designed inputs, which encode information from a pre-defined window of frames into a single RGB image, the architectures proposed in this Chapter attain the final purpose of single-frame classification. As each model will be classified based only on partial observations of each action, this falls over the category of HAP (see Section 2.2), more specifically tackling early action recognition and detection.

4.1 Approaches

Tackling human motion decoding, through raw RGB data from a SW's moving camera, is a challenging new problem. Therefore, different model architectures and frameworks should be first explored and compared, in order to effectively infer the best way to solve it or at least the direction to follow towards improvements.

The devised approaches are clarified in Figure 4.1, which integrated different model architectures, each one fed with all the computed forms of input. Brief summaries of these approaches are presented below:

- **Approach 1:** The different computed inputs were fed into the baseline CNN classification models and their performances evaluated, aiming the perception of the most suitable CNN architecture for

this task. Details about the framework and model architectures used in this approach are depicted in Section 4.2.

- **Approach 2:** An attention mechanism was added to the selected CNN architecture and then trained and evaluated on each one of the input forms. Details of this approach, including the attention mechanism's architecture, are also described in Section 4.2, as these two first approaches share the same overall framework.

- **Approach 3:** All the input forms were also tested in a segmentation-classification framework, described in Section 4.3.

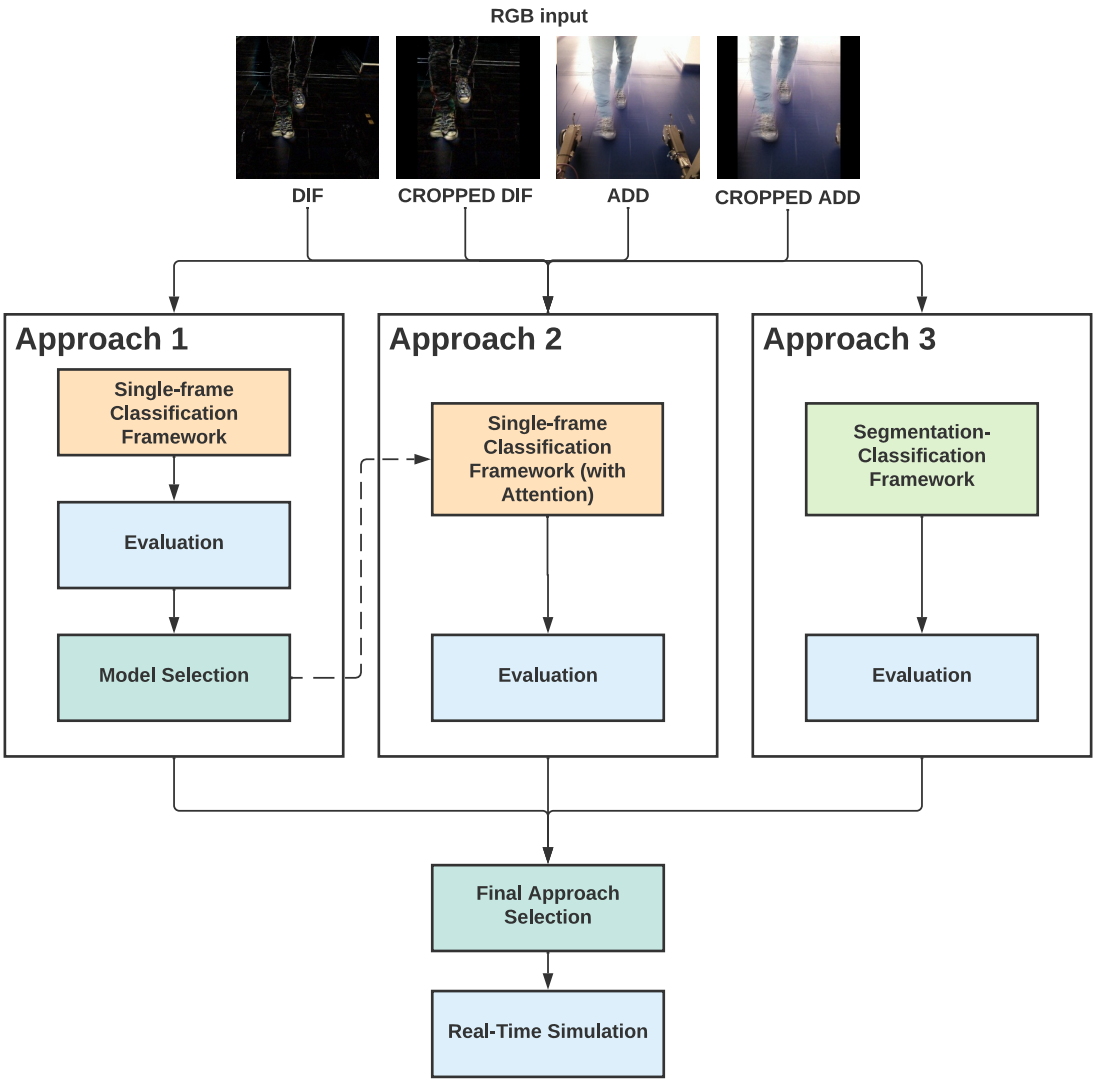


Figure 4.1: Overall flowchart of the work developed, depicting all the different approaches.

The evaluation results are addressed in Chapter 5, following this architectures' order.

4.2 Single-frame Classification Framework

The framework illustrated in Figure 4.2 was implemented to perform the first two previously depicted approaches. Here, the final inputs are generated from raw RGB data, following the preprocessing procedures explained in Section 3.4.2. Then, the models are trained and tested for the classification of each form of input, generating a final output vector of length 4 (matching the number of target classes).

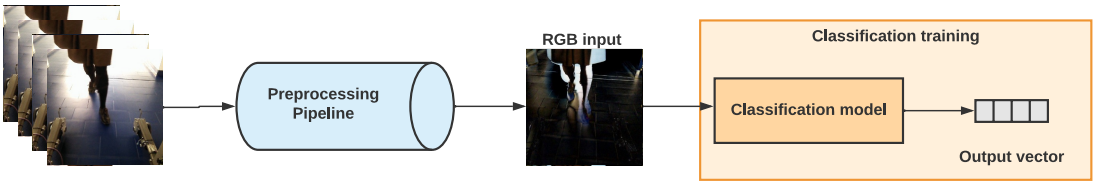


Figure 4.2: Schematic of the single-frame classification framework.

According to Figure 4.1, the models used in this framework correspond to: **approach 1)** the baseline CNN models (VGG16 and ResNet-50) and **approach 2)** the chosen baseline with an attention mechanism. These models' architectures are described in the following subsections.

4.2.1 Baseline Architectures

The first model employed was VGG16 [57], also recently used in [29] for HAR. As in [29], here the VGG16 model was used without its top layers. Instead, the extracted feature maps are fed to a Global Average Pooling (GAP) layer, followed by a softmax layer with four units, as illustrated in Figure 4.3. Additionally, two more alterations were introduced: **i)** the activation function of the last convolutional layer was changed from Rectified Linear Unit (ReLU) to *tanh* function, to decrease the clipping of final feature maps' values and avoid possible vanishing gradients, when using units; **ii)** BatchNorm layers were added, to improve the optimisation and generalisation performance, after the activation layer. This differs from the most common practice, where BatchNorm layers are added before the activation.Nonetheless, the difference in these layers' positions should not imply significant changes in the results. Also, some researches defend the architecture used here [82].

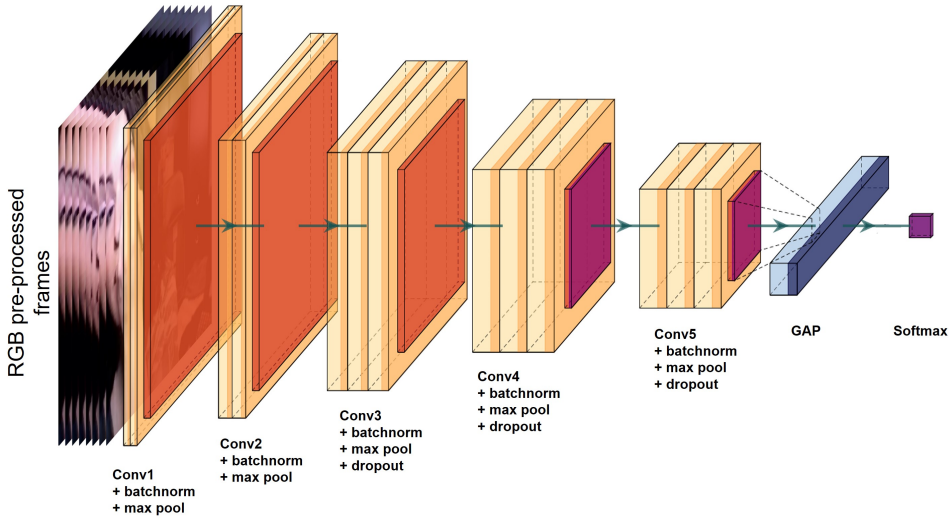


Figure 4.3: Diagram of the implemented VGG16 architecture.

The second CNN model was ResNet-50 [58], also followed by a GAP and softmax (Figure 4.4).

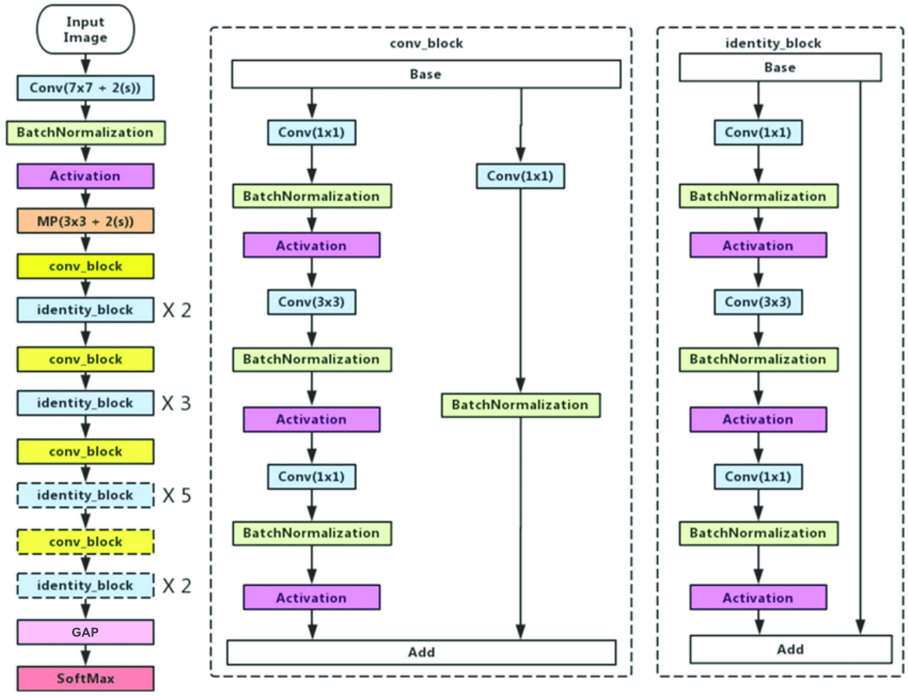


Figure 4.4: Schematic of the implemented ResNet-50 model: (**Left**) Model architecture (MP stands for MaxPooling). (**Middle**) Architecture of the convolution block which changes the dimension of the input. (**Right**) Architecture of the identity block which will not change the dimension of the input. This image was retrieved from [83].

4.2.2 Attention Mechanism

It has been defended that the different channels of CNN convolutional features correspond to different feature detectors and thus ignoring their distinct learning abilities may lead to a decrease in CNN

performance. Therefore, inspired by [70], an attention mechanism was attached to the chosen baseline model (selected in Chapter 5), aiming to automatically learn these channel-wise features, while adaptively enhancing the informative channels by assigning different weights to each channel. This attention mechanism, whose architecture can be seen in Figure 4.5, follows the CNN model's last convolutional block. Its outputted feature maps are used by the *Channel Attention Module* to compute weights for each output channel, forming a channel descriptor vector. A multiplication is then performed between each feature map and the corresponding weight of the computed vector, generating the final *Channel-attention Weighted Feature Maps*. Finally, these are fed to the GAP layer of the selected baseline CNN model. The selection of this baseline architecture will be detailed along Chapter 5.

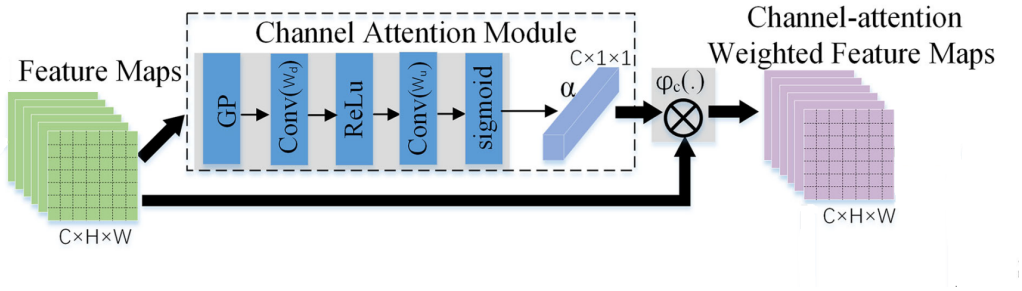


Figure 4.5: Diagram of the implemented channel-wise attention mechanism, retrieved from [70]. Here, the feature maps correspond to the last convolutional feature maps of the ResNet-50 model.

4.3 Segmentation-Classification Framework

After evaluating the model's focus and relevance of the extracted features, a novel framework was designed as an attempt to drive the model's focus to the relevant areas of the RGB input, mainly the human body. Based on studies that treat segmentation and classification as inter-dependent tasks [84], a two-stage framework was developed, targeting the execution of **approach 3**: **Stage 1)** a segmentation model is trained with the preprocessed RGB frames and the corresponding computed masks; **Stage 2)** the same type of input is fed into a classification network, pre-trained with the learned weights from Stage 1.

A visualisation of this framework is shown in Figure 4.6, where the referred GT masks are pre-computed through an algorithm described in Section 3.4.3.

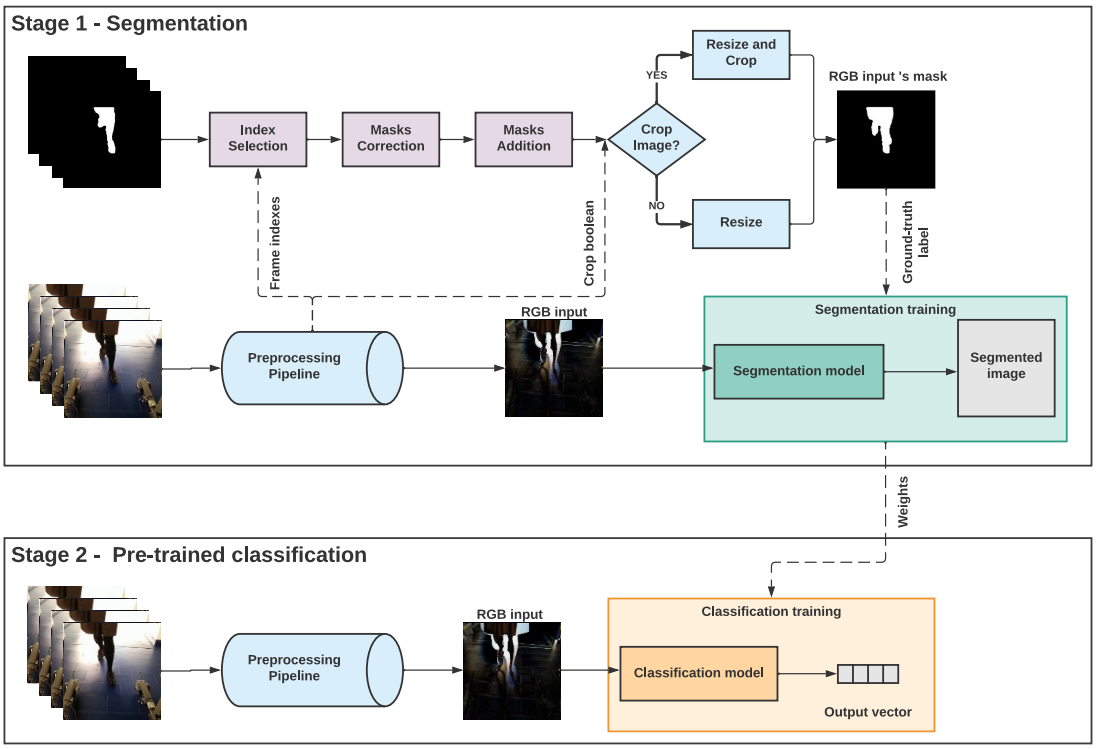


Figure 4.6: Schematic of the segmentation-classification framework.

The implemented DL architectures are detailed in Section 4.3.1, as follows.

4.3.1 Segmentation-Classification Architectures

The model chosen to segment single RGB preprocessed images was the UNET model [85], with additional BatchNorm layers, as previously described in the VGG16 architecture. Figure 4.7 illustrates the respective architecture. The input frames have dimension of 224x224, so the lowest resolution achieved is 14x14. The last convolutional layer (1x1) uses sigmoid activation and outputs a 224x224 segmentation map.

From this segmentation architecture, an adapted UNET model for classification was designed, integrating the UNET's encoder followed by two convolutional blocks, as shown in Figure 4.8. BatchNorm layers were also added to the UNET encoder.

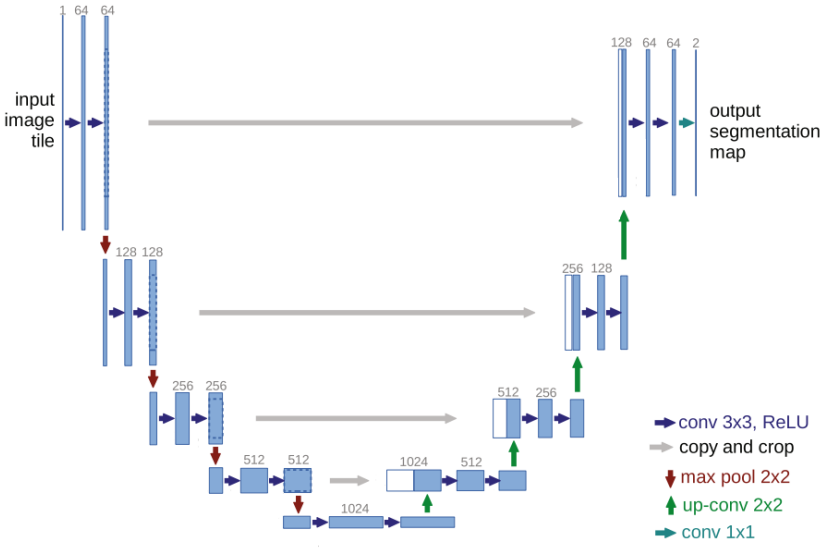


Figure 4.7: Schematic of U-net architecture, where each blue box corresponds to a multi-channel feature map, with its number of channels annotated on its top, the white boxes represent copied feature maps and different operations are denoted by arrows. Retrieved from [85].

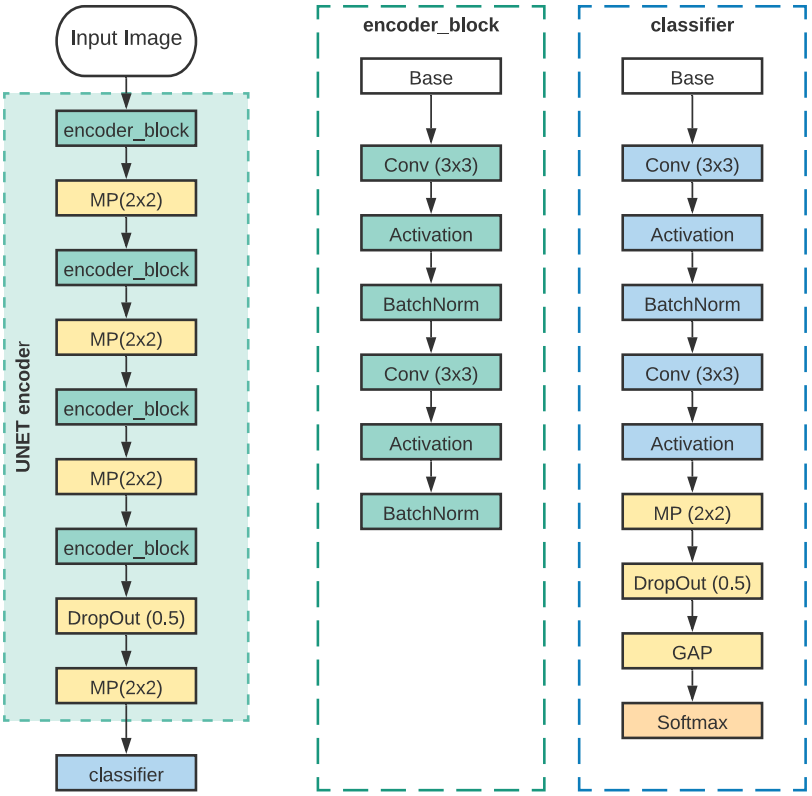


Figure 4.8: Schematic of the adapted UNET model for single-frame classification (MP stands for MaxPooling).

4.4 Real-Time Simulation

After running the three approaches in Figure 4.1, all the obtained results were compared. The best framework and model architecture were selected and evaluated in real-time simulations, assessing the performance in OAD and early action detection tasks.

4.4.1 Post-processing

As the future purpose of this dissertation will be the model deployment on a SW prototype, a post-processing technique was developed in order to optimise the performance in a real-time environment. Its scope consisted on dealing with the model's uncertainty, while reducing possible on-off noise, without introducing significant delays in the decision process.

This post-processing technique sets a minimum action duration (2s) and applies it to the previous predicted class, meaning that it only allows for a transition to happen, if the previous predicted action lasted at least 2s. Additionally, it also reduces the spectrum of possible transitions by not allowing a turn right/left to happen right after a turn left/right, respectively. These conditions help avoiding prediction errors and are consistent with rehabilitation therapy sessions.

4.5 Experimental Protocol

All the implementation details involved in the dataset partition (Section 4.5.1), as well as the actual training (Section 4.5.2) and evaluation (Section 4.5.3) of the proposed models are described in the following sections. Additionally, it details the training hardware used throughout the development of this dissertation.

4.5.1 Dataset Split

Due to limitations of time and computational resources, the models were evaluated not through cross-validation, but through a default and constant dataset splitting. More specifically, 20% was used for testing (3 subjects) and 80% for training (12 subjects), where one subject was left to the validation set [41]. Random subjects were chosen, as they are all adults with healthy gait patterns. It was divided by subjects to guarantee a good distribution and no overfitting. Table 4.1 describes each of these splits, already considering the inputs computed through a sliding window of length 4 (Section 3.4.1) over the created balanced dataset of frames (Section 3.3.2). For the tasks involving masks (segmentation and

grad-CAMs evaluation), the dataset used was smaller, since some sequences were removed for mask corruption reasons. More precisely: 171 sequences in the train set (32%), 13 in the validation set (27%) and 16 in the test set (11%). These numbers correspond to the dataset used for offline training and evaluation.

The test dataset used for real-time simulations (Section 5.3) is composed by the same subjects present in the depicted test split (Table 4.1). Although, this dataset is composed by the whole trials, instead of being divided into 40-frame sequences.

Table 4.1: Constitution of each dataset split, containing the inputs computed from a sliding window of 4 frames

Split	Subjects IDs	Number of images	
		Classification	Segmentation
Train	$[1,15[\setminus \{5, 8, 11\}$	19536	13209
Validation	5	1776	1295
Test	8, 11, 15	5328	4736

4.5.2 Implementation Details

All the models used in this project were developed offline, with the collected data, using the Tensorflow and Keras DL library on a Python environment. The DL models were trained, in most of this work, on a free Google Colab instance [1] with the following hardware specifications: **GPU**: 1x Nvidia Tesla T4 (16GB VRAM); **CPU**: Xeon Processor @2.3GHz (1 core, 2 threads); **RAM**: 12.6 GB; **Disk**: 33 GB; This instance is also limited to 8H of continuous use, time after which it gets recycled. In the final stages, final experiments and approaches were trained in a computer with the following hardware specifications: **GPU**: 1x Nvidia GeForce RTX 3080 Ti/PCIe/SSE2; **CPU**: Intel(R) Core(TM) i9-10940X @3.3GHz (14 core, 28 threads); **RAM**: 65,5 GB.

Reasonable hyperparameters were extracted from literature that tackles similar tasks and models ([14] for single-frame classification and [85] for segmentation). The following tables denote the hyperparameters used for segmentation (Table 4.2) and single-frame classification (Table 4.3). The initial learning rate is decayed by 50% until a minimum of $1e^{-4}$, if the training loss does not improve within 4 epochs. At the end of the training, the best model was selected according to the validation f1-score or loss, for classification and segmentation, respectively. Additionally, the training finishes earlier if these metrics stop improving, after 30 epochs.

[1] https://colab.research.google.com/

Table 4.2: Hyperparameters defined for the developed single-frame segmentation algorithms

	Parameters
Window Length	4 frames
Loss Function	Binary Cross-Entropy
Optimiser	Adam
Batch size	16
Epochs (shuffled)	100
Initialiaser	He normal
Learning rate	$1e^{-4}$
Data Augmentation	None
Callbacks	Checkpoint and Early Stopping (metric = val_loss)

Table 4.3: Hyperparameters defined for the developed single-frame classification algorithms

	Parameters
Window Length	4 frames
Loss Function	Categorical Cross-Entropy
Optimiser	Mini Batch Gradient Descent with Nesterov momentum
Batch size	64
Epochs (shuffled)	100
Number of frozen layers*	16
Learning rate - Momentum	0.001 - 0.9
Data Augmentation	Width and Height shifts = 15, Zoom = 0.2, Brightness = [0.75,1.25], Contrast = [0.75,1.25], Gaussian Blur = [(3,3), (5,5)]
Callbacks	Checkpoint, Reduce LR on Plateau and Early Stopping (metric = val_f1_score)

* only applicable in the segmentation-classification approach

Table 4.4 presents a brief summary of the number of parameters for each one of the proposed and trained models.

Table 4.4: Number of parameters for each trained model

Model	Total Parameters	Trainable Parameters	Non-trainable Parameters
VGG16	14 731 588	14 724 164	7 424
ResNet-50	23 595 908	23 542 788	53 120
ResNet50 + attention	25 695 620	25 642 500	53 120
UNET	31 060 237	31 046 537	13 700
Adapted UNET	6 465 092	5 313 540	1 151 552

Transfer Learning

Facing small-sized training dataset problems, and specially its low variability caused by the cyclical nature of the gait, transfer learning techniques were implemented, in order to improve performance. As [29], the backbone weights of VGG16 and ResNet-50 classification models (see Section 4.2) were initialised to the weights of these same state-of-the-art networks pre-trained on the general object classification benchmark, ImageNet [86], towards accuracy maximisation, while using large amounts of computational resources, which trains the models to detect and produce general features.

In the segmentation-classification approach, the UNET encoder's weights of the classification model were initialised with the ones learned in the previous training of the UNET model for segmentation. The classifier's convolutional layers were here initialised with the He normal function.

4.5.3 Key-Performance Indicators

Classification

According to the literature, the models were evaluated using common metrics for classification, namely Accuracy, Precision, Recall and F1-score [29] [14]. These metrics are defined in the equations below, including True Positives (TP), True Negatives (TN), FP, FN and n, which stands for the total number of classes (in this case, 4).

$$\textbf{Macro Accuracy} = \sum_{i=1}^{n} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{4.1}$$

$$\textbf{Macro Precision} = \frac{1}{n} \sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i} \tag{4.2}$$

$$\textbf{Macro Recall} = \frac{1}{n} \sum_{i=1}^{n} \frac{TP_i}{TP_i + FN_i} \tag{4.3}$$

$$\textbf{Macro F1-score} = \frac{2 * MacroPrecision \times MacroRecall}{MacroPrecision + MacroRecall} \tag{4.4}$$

Segmentation and Grad-CAMs evaluation

For segmentation and grad-CAMs quantitative evaluation, the metrics were computed between each segmented image or grad-CAM heatmap (generated through the algorithm described in [31]) and the respective GT mask. Mean Intersection over Union and Dice are common evaluation metrics for semantic image segmentation and the respective formulas are shown below, where *n* here equals to 2 classes, one for the background and another for the foreground.

$$\textbf{Mean IoU} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{4.5}$$

$$\textbf{Mean Dice} = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times TP_i}{(TP_i + FP_i) + (TP_i + FN_i)} \tag{4.6}$$

Note that the grad-CAMs are computed over the model's last convolutional layer, following [31], and their evaluation algorithm computes the mean metrics over all the inputs of the dataset. This evaluation is performed only on the validation and test sets.

Real-time (Online) simulations

Inspired by [32], OAD metrics were developed to better evaluate the performance of the real-time simulations, such as IA, IP, wIA and cIP. These evaluate the model's performance as the frames are acquired, without having to wait to an unknown end. Additionally, wIA and gIscIP conditions the value of a true observation (TP and TN) to the total negatives *vs.* total positives ratio (w), which is dynamic and always based only on the seen portion of the video. These are represented in the following equations, were t corresponds to the time instant, NC to the number of classes and TP, TN, FP and FN refer to the seen true/false positive/negative observations overall classes.

$$\textbf{IA} = \sum_{j=1}^{t}\frac{TP_j + TN_j}{t \times NC} \tag{4.7}$$

$$\textbf{IP} = \frac{\sum_{j=1}^{t} TP_j}{\sum_{j=1}^{t} TP_j + FP_j} \tag{4.8}$$

$$\textbf{wIA} = \frac{w \times \sum_{j=1}^{t} TP_j + \frac{1}{w} \times \sum_{j=1}^{t} TN_j}{t \times NC} \tag{4.9}$$

$$\textbf{cIP} = \frac{w \times \sum_{j=1}^{t} TP_j}{w \times \sum_{j=1}^{t} TP_j + \sum_{j=1}^{t} FP_j} \tag{4.10}$$

Chapter 5

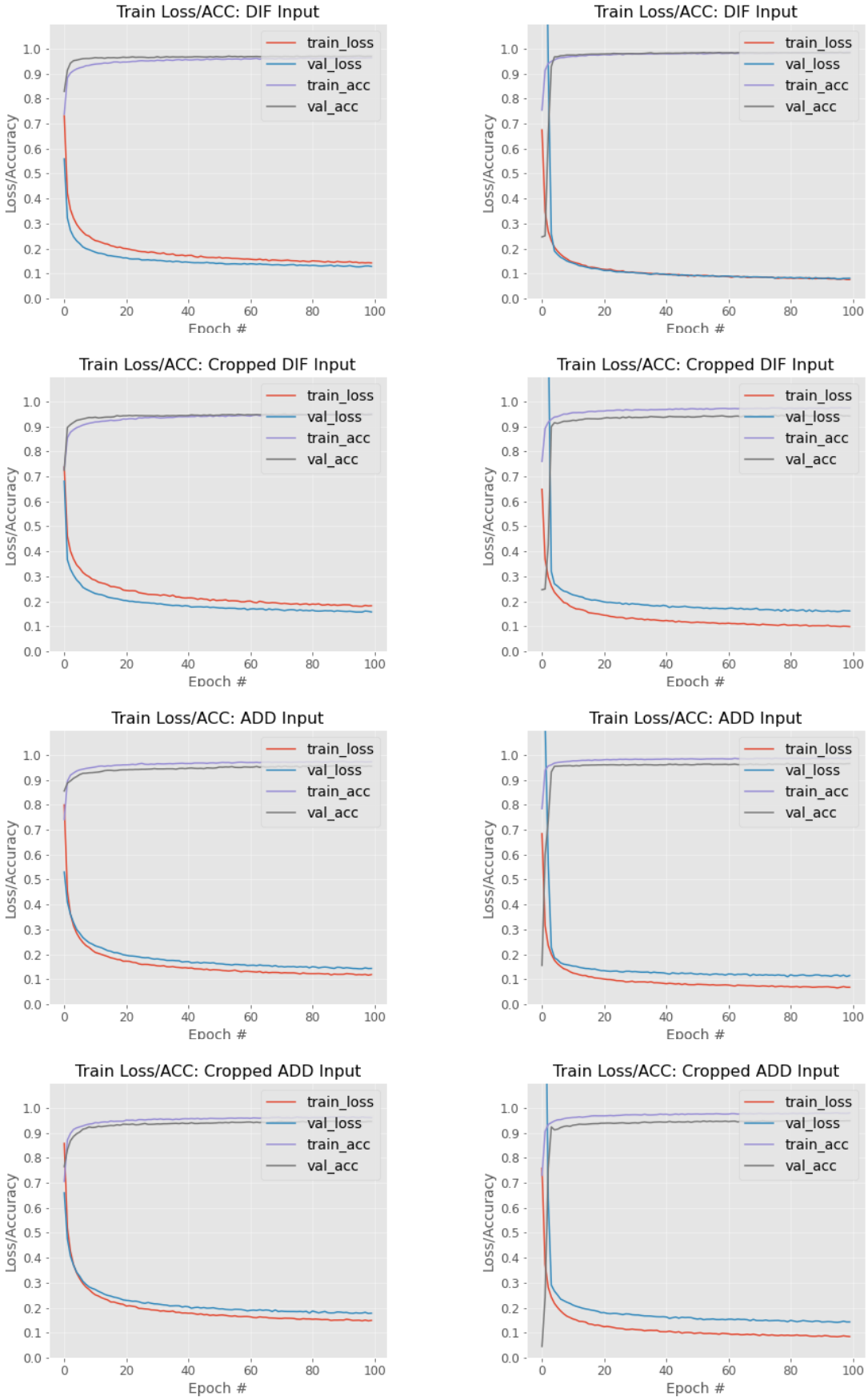
Results

The proposed approaches are evaluated in the following sections, starting from **approach 1:** baseline CNN models (Section 5.1.1) and **approach 2:** the attention mechanism (Section 5.1.2) for single-frame classification; to **approach 3:** the segmentation-classification framework (Section 5.2). The influence of various types of input is studied and the classification models evaluated in two aspects: **i)** the accuracy of the predicted labels; **ii)** the grad-CAMs' heatmaps similarity with the GT masks. Although the common practice dictates that test results should not be considered in the choice of the best approach, in this work the validation set has less variability and samples than the test set and so, to build up a more consistent opinion, both datasets were considered in the following evaluations of each model.

5.1 Single-frame Classification Framework

5.1.1 Baseline Models

Training VGG16 and ResNet-50 architectures (Section 4.2.1) with each developed input, computed from the acquired dataset, resulted in the training curves presented in Figure 5.1. As one can see, the overall curves are stable and with no signs of overfitting, reaching good results. It is noticeable that ResNet-50 learned faster and provided some gains in loss and accuracy.



(a) VGG16. **(b)** ResNet-50.

Figure 5.1: Accuracy and loss training curves for VGG16 and ResNet-50 models.

Table 5.1 presents the validation results, as well as the training time, of the VGG16 (Figure 4.3) and ResNet-50 (Figure 4.4) classification models for each type of input (*i) DIF*, *ii)* cropped *DIF*, *iii) ADD* and *iv)* cropped *ADD*). All the models were trained in 100 epochs.

Table 5.1: Validation results of the VGG16 and ResNet-50, as well as the training time for 100 epochs

Input Type	Crop	ACC (%)	Loss	F1-score (%)	Precision (%)	Recall (%)	Training Time (h)
VGG16							
DIF	False	**97.02**	**0.13**	**96.80**	97.38	**96.23**	4.45
DIF	True	94.76	0.16	95.02	95.66	94.37	4.47
ADD	False	95.50	0.14	96.28	**97.79**	94.82	**4.44**
ADD	True	94.48	0.18	94.53	94.99	94.03	4.54
ResNet-50							
DIF	False	**98.42**	**0.08**	**98.27**	**98.52**	**98.03**	4.45
DIF	True	94.37	0.16	94.34	95.24	93.47	4.47
ADD	False	96.34	0.12	96.26	96.65	95.83	**4.44**
ADD	True	94.87	0.14	95.76	97.05	94.48	4.46

Concerning the classification task, it is evident that the ResNet-50 model outperforms the VGG16, except for the cropped *DIF* input. Cropping the images interferes with the model's outcomes, increasing the loss and the training time, while decreasing the remaining metrics. The *DIF* revealed here a better performance, followed by the *ADD* input, which was only worst by an overall maximum margin of approximately 2%.

The test results are illustrated by the confusion matrices shown in Figures 5.2 and 5.3. Table 5.2 reveals the percentage of wrongly classified test frames.

Table 5.2: Percentage of wrongly classified frames in the test set, by the VGG16 and ResNet-50 models

Input Type	Crop	VGG16	ResNet-50
DIF	False	3.79%	2.06%
DIF	True	6.31%	4.97%
ADD	False	**2.49%**	**1.16%**
ADD	True	4.34%	3.57%

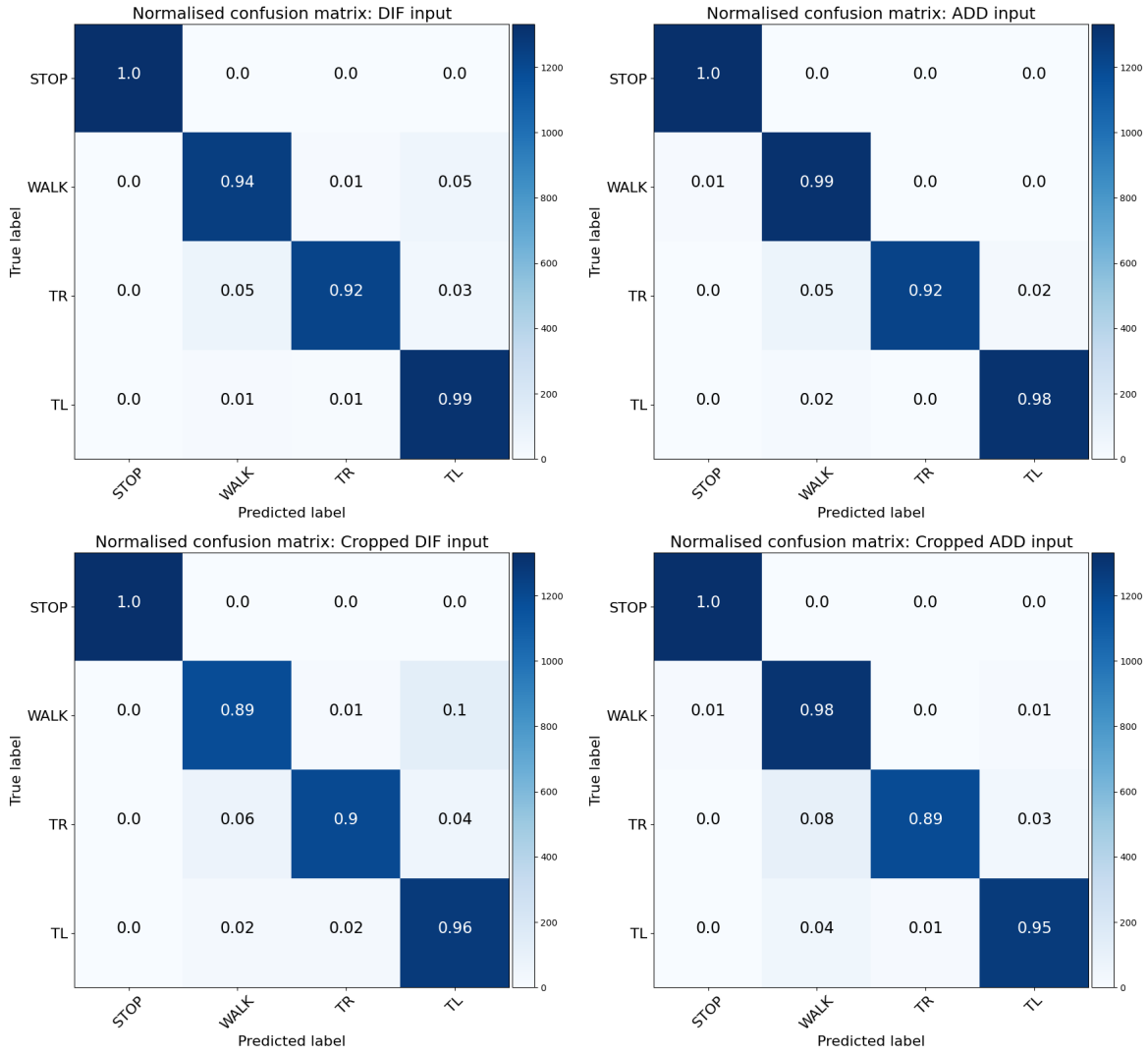


Figure 5.2: Confusion matrices for the VGG16 model over the four types of input.

According to Table 5.2, the higher ability for correctly classifying the test set images was registered by the ResNet-50 model and the *ADD* input was easier to classify for both models (2.49% and 1.16% of wrongly classified *ADD* test images, for VGG16 and ResNet-50, respectively). Once again, the cropped inputs showed worse performances, which can also be inferred by the comparison of the confusion matrices. Nonetheless, these matrices show very good results, overall classes and inputs.

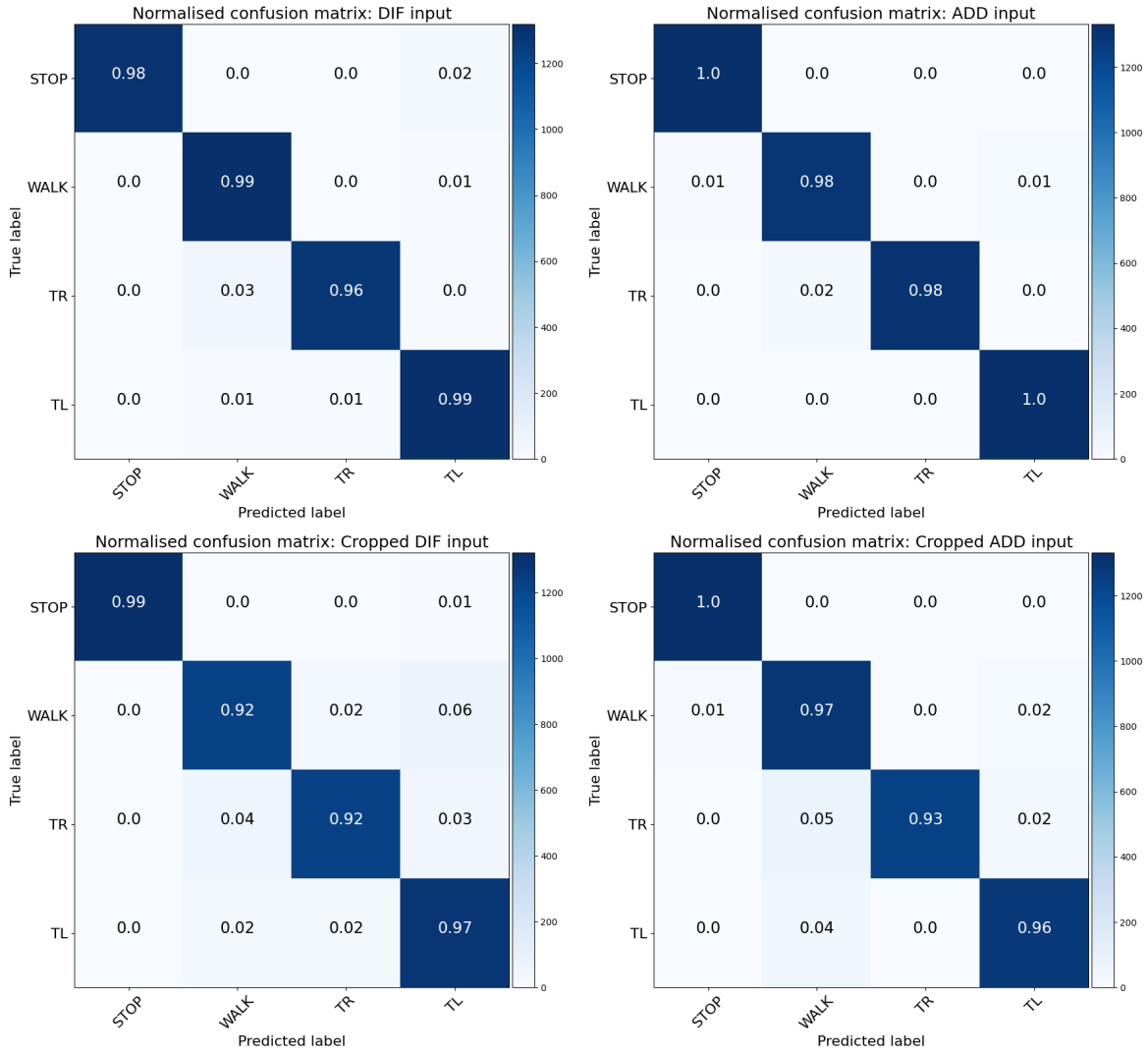


Figure 5.3: Confusion matrices for the ResNet-50 model over the four types of input.

When evaluating the grad-CAMs focus, for the validation and test subsets, the following results were obtained (Table 5.3).

Table 5.3: Quantitative evaluation results of the validation and test grad-CAMs, when predicting with the VGG16 and ResNet-50 models

		Validation				Test			
Input		**VGG16**		**ResNet-50**		**VGG16**		**ResNet-50**	
Type	**Crop**	**Mean Dice (%)**	**Mean IoU (%)**	**Mean Dice (%)**	**Mean IoU (%)**	**Mean Dice (%)**	**Mean IoU (%)**	**Mean Dice (%)**	**Mean IoU (%)**
DIF	False	16.16	9.10	29.06	17.70	17.18	9.64	26.01	15.84
DIF	True	22.57	13.44	32.09	19.67	26.39	15.91	28.38	17.21
ADD	False	20.19	11.4	22.09	13.36	20.99	11.94	16.93	9.95
ADD	True	**28.34**	**16.84**	**32.13**	**19.89**	**28.62**	**17.18**	**29.37**	**17.86**

Contrarily to the quantitative classification results, here the cropped inputs present better focus, mean-

ing a higher similarity between the model's grad-CAMs and the GT human body masks. In average, cropping part of the background from the inputs increased these Mean Dice and Mean IoU metrics by 5.26% and 2.50%, for the *DIF* input, and by 9.57% and 6.28%, for the *ADD* input, respectively. The cropped *ADD* input revealed better grad-CAMs focus, achieving its best results with the ResNet-50 model. This model showed an overall average boost of 17.82% in Mean Dice and 13.02% in Mean IoU metric, when compared to the VGG16.

5.1.2 Attention Mechanism

Taking into consideration the superior results obtained by the ResNet-50 model (5.1.1), a channel-wise attention mechanism was added to this model, according to **approach 2** (Figure 4.5), and its influence is evaluated in this Section. Figure 5.4 shows the obtained training curves.

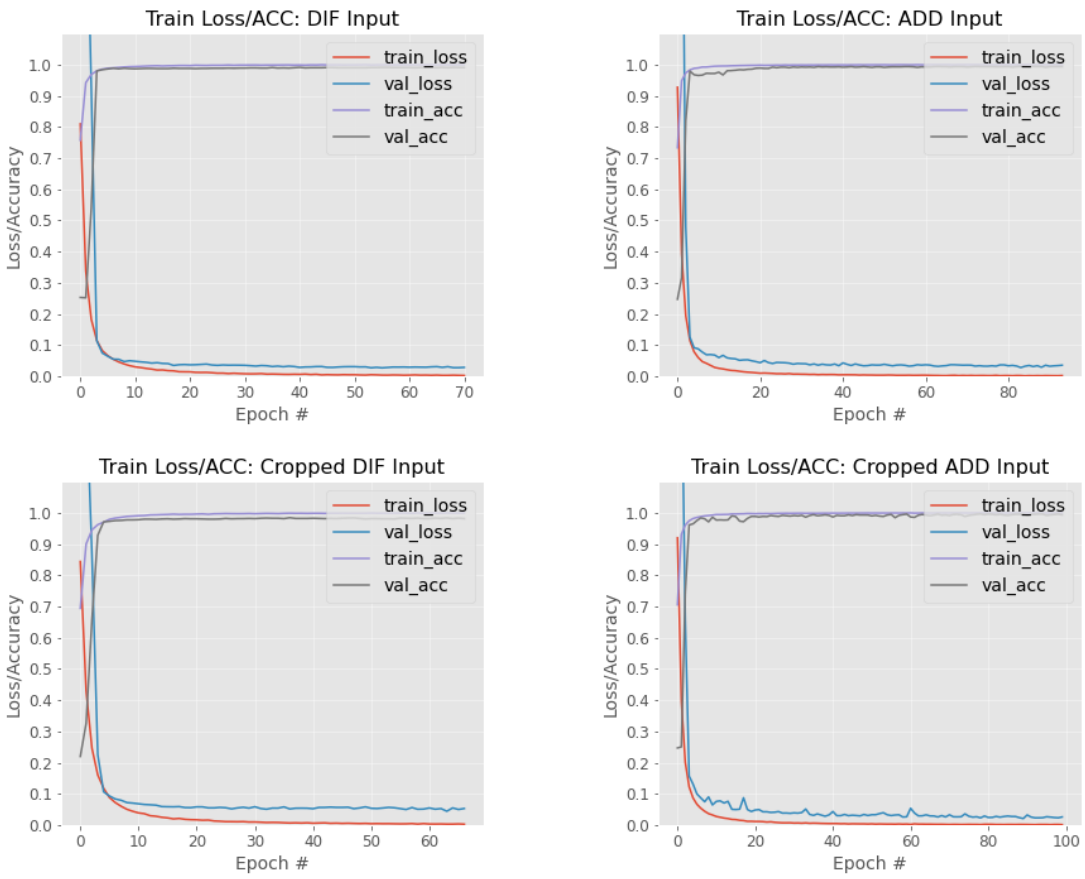


Figure 5.4: Accuracy and loss training curves for ResNet-50 model with an attention mechanism.

Table 5.4 presents the results achieved by this model, when evaluated in the validation set, as well as the training time and number of epochs.

Table 5.4: Validation results of the ResNet-50 model with a channel-wise attention mechanism, as well as the training time and number of epochs

Input Type	Crop	ACC (%)	Loss	F1-Score	Precision (%)	Recall (%)	Number of epochs	Training Time (h)
DIF	False	99.04	0.03	99.03	99.04	99.04	71	2.93
DIF	True	98.31	0.05	98.32	98.37	98.25	67	2.76
ADD	False	99.38	0.03	99.39	99.38	99.38	94	3.86
ADD	True	**99.61**	0.03	**99.61**	**99.61**	**99.61**	100	4.17

It can be seen that these results show a significant improvement compared to the baseline ResNet-50 model (Table 5.1), specially for the cropped inputs and even when training with less epochs. Adding the channel-wise attention mechanism increased the f1-score by 3.98% and 3.85%, for the cropped *DIF* and *ADD* inputs, respectively. Note that the different number of epochs presented in Table 5.4 (from 67 to 100 epochs) is due to the early stop of the training, when the validation f1-score stopped improving.

Comparing Table 5.5 with Table 5.2, it is also noticeable that this model enhanced the classification performance over the unseen test set. With attention, the percentage of wrong predictions by the ResNet-50 model decreased from 3.57% to 0.64%, for the cropped *ADD* input. Once again, the *ADD* input type stood out, not only in the test results (0.71% and 0.64%), but in validation as well (f1-score of 99.39% and 99.61%, for non-cropped and cropped inputs, respectively).

Table 5.5: Percentage of wrongly classified frames in the test set, by the ResNet-50 model with an attention mechanism

Input Type	Crop	Wrong predictions (%)
DIF	False	0.68
DIF	True	2.38
ADD	False	0.71
ADD	True	**0.64**

Figure 5.5 presents the confusion matrices obtained for the test set. These matrices reveal excellent results, specially for the STOP and TL classes, where the TP rate was never lower than 0.99. The cropped *DIF* was the least favoured input by this architecture, with the lowest TP rate for the WALK class (0.93).

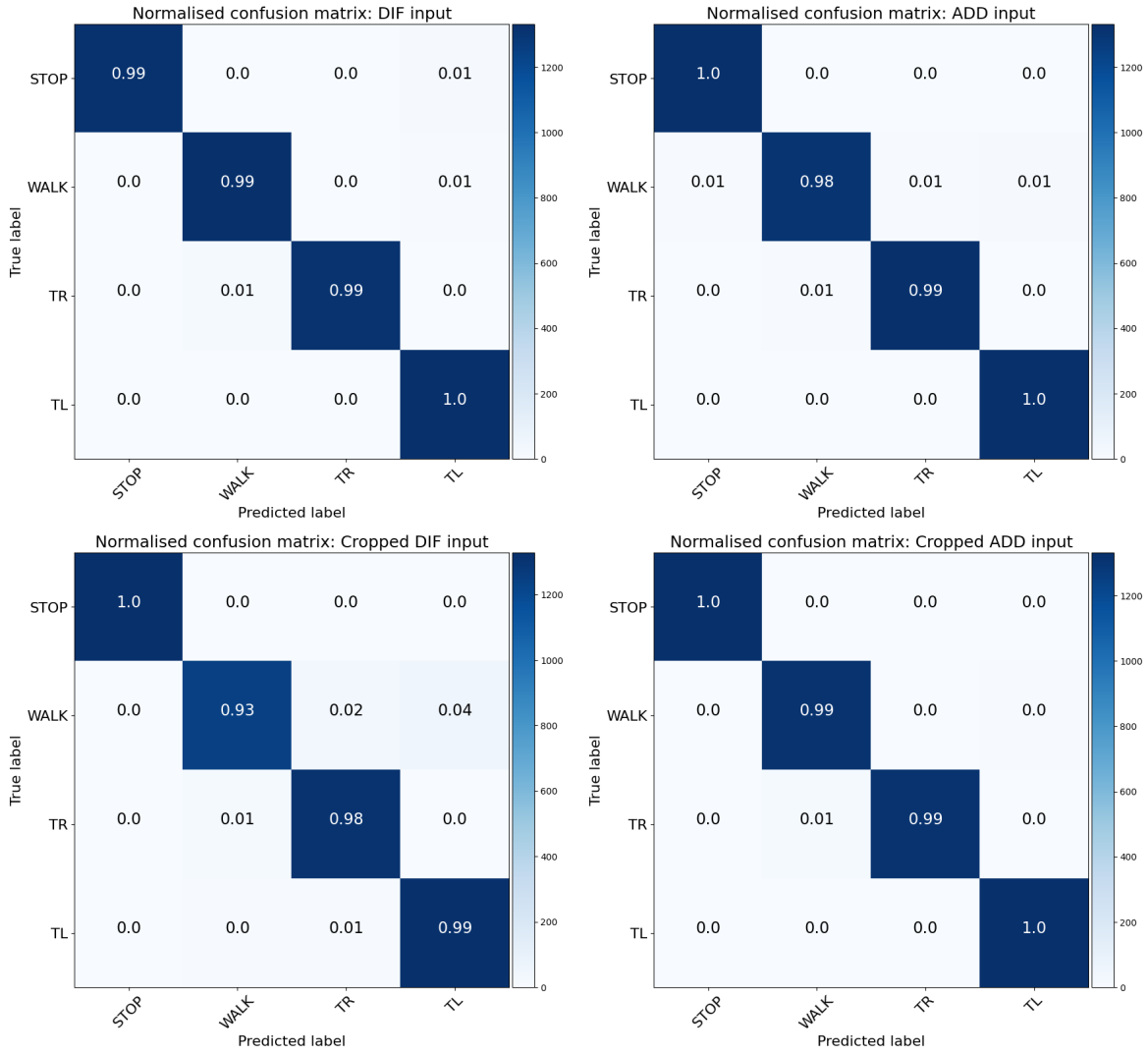


Figure 5.5: Confusion matrices for the ResNet-50 model with attention, over the four types of input.

Table 5.6 reveals the results obtained from the evaluation of this model's focus. When comparing them with the baseline model's results (Table 5.3), it is noticeable that the attention mechanism improves, even if just for a little margin (< 5.3%), the focus of the ResNet-50 for every input. The greatest improvement was recorded by the non-cropped *ADD*, corresponding to 4.21% and 5.21% in Mean Dice values, for validation and test, respectively. However, this one still lead to lower metrics than the non-cropped *DIF* input, as the latter achieved Mean Dice values higher by 4.07% (validation) and 7.83% (test) than the non-cropped *ADD*. Contrarily to previous results, cropped *DIF* attained the higher similarity between their GT masks and the grad-CAMs obtained from the ResNet-50 model with attention. Nonetheless, the difference relative to the cropped *ADD* is not significant (not higher than 1.11% in Mean Dice).

Table 5.6: Quantitative evaluation results of the validation and test grad-CAMs, when predicting with the ResNet-50 model with an attention mechanism

Input		Validation		Test	
Type	**Crop**	**Mean Dice (%)**	**Mean IoU (%)**	**Mean Dice (%)**	**Mean IoU (%)**
DIF	False	30.37	18.59	29.97	18.50
DIF	True	**32.90**	**20.04**	**32.38**	**19.76**
ADD	False	26.30	15.99	22.14	13.07
ADD	True	31.79	19.21	32.30	19.60

5.2 Segmentation-Classification Framework

5.2.1 Segmentation

UNET model's segmentation power over the different forms of input proved itself sufficiently good for its purpose in this dissertation, which is initialising the weights of its encoder for a classification task, in order to study if this could lead to more human-centred extracted features. The results confirming this statement are shown in Table 5.7, where the presented values of Mean IoU are in line with the ones found in the literature [84].

Using weights pre-trained with the acquired WALKit dataset to train the classification model can already induce a lower power of generalisation. As shown in Figure 5.6, the training was shortened to 30 epochs, for every input, since the segmentation revealed itself as an easy task, prompt to a little overfitting in a longer training, which could compromise even more the generalisation ability of the following classification model (Section 5.2.2).

Table 5.7: Validation results of the UNET model, as well as the training time for 30 epochs

Input Type	Crop	ACC (%)	Loss	Mean IoU (%)	Mean Dice (%)	Training Time (h)
DIF	False	**98.80**	0.02	44.74	95.17	1.37
DIF	True	98.36	0.03	42.03	95.51	1.37
ADD	False	98.94	0.02	**86.39**	**96.23**	1.38
ADD	True	98.57	0.03	41.83	96.18	1.37

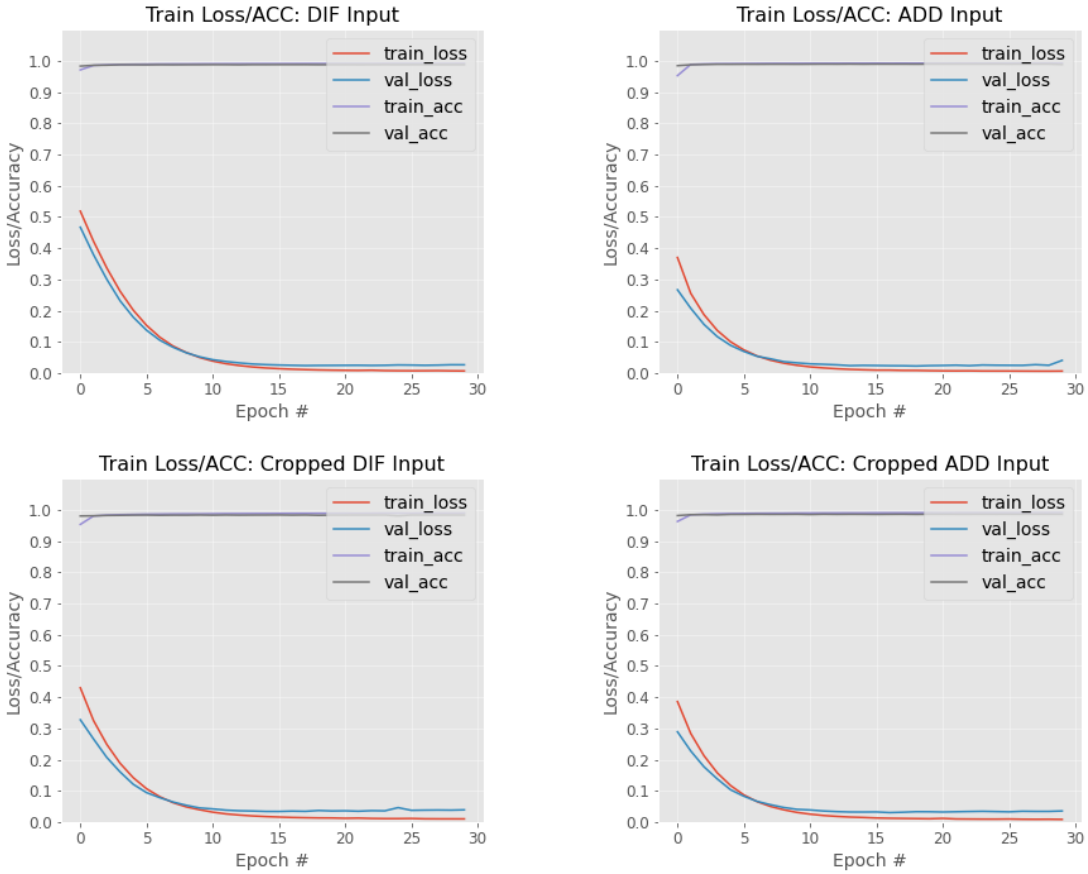
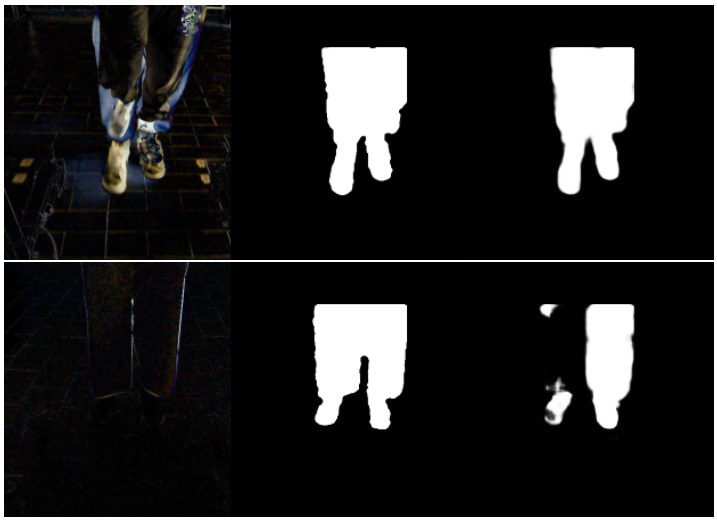


Figure 5.6: Accuracy and loss training curves for segmentation.

To help visualise this model's segmentation ability, Figures 5.7, 5.8, 5.9 and 5.10 show the best and worst cases of segmented test images, for each type of input, based on computed similarity values between the GT mask and the respective segmented image. Note that, for the cropped *ADD*, the segmentation of the human body was very satisfactory, even in the worst case, although including some noise. Contrarily to this, the other inputs revealed occlusions as the apparent main factor behind a worse segmentation. The quantitative test results shown in Table 5.8 indicate that the cropped *ADD* images were better segmented by this model, followed by the non-cropped *ADD*, cropped *DIF* and, finally, non-cropped *DIF* images.

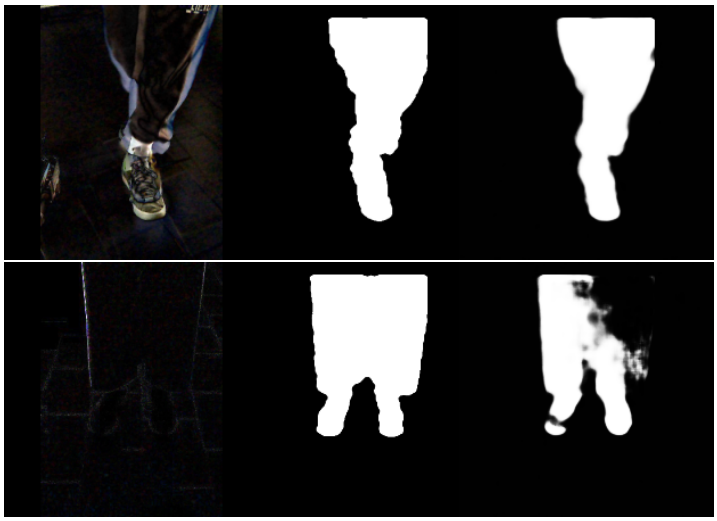
Table 5.8: Evaluation results of the UNET segmentation model over the test set

Input Type	Crop	Mean IoU (%)	Mean Dice (%)
DIF	False	88.43	93.74
DIF	True	89.95	94.65
ADD	False	90.83	95.09
ADD	True	**92.15**	**95.88**



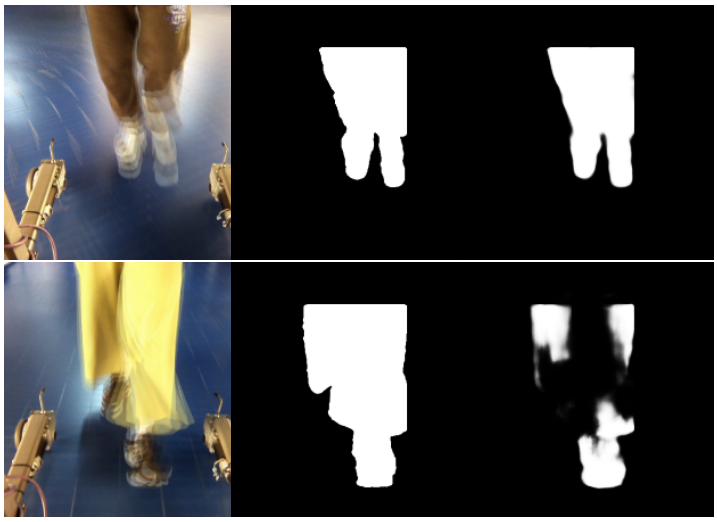
(a) RGB input. **(b)** GT mask. **(c)** Segmented image.

Figure 5.7: Examples of the best (upper) and worst (lower row) cases of segmented images, along with the respective non-cropped *DIF* inputs and labels.



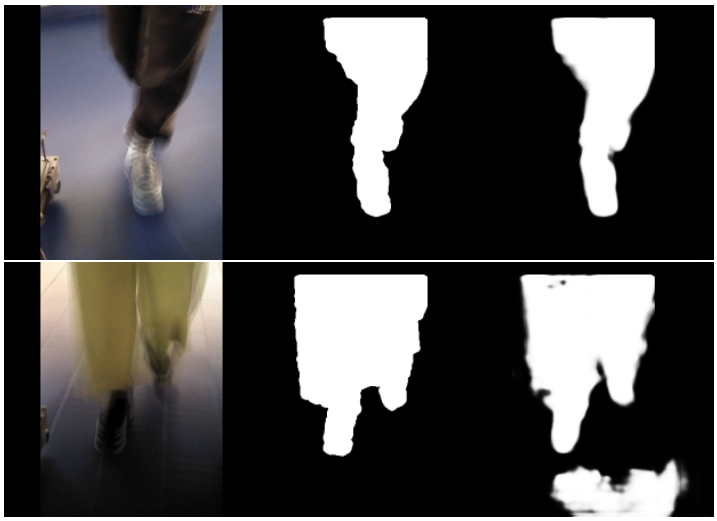
(a) RGB input. **(b)** GT mask. **(c)** Segmented image.

Figure 5.8: Examples of the best (upper) and worst (lower row) cases of segmented images, along with the respective cropped *DIF* inputs and labels.



(a) RGB input. **(b)** GT mask. **(c)** Segmented image.

Figure 5.9: Examples of the best (upper) and worst (lower row) cases of segmented images, along with the respective non-cropped *ADD* inputs and labels.



(a) RGB input. **(b)** GT mask. **(c)** Segmented image.

Figure 5.10: Examples of the best (upper) and worst (lower row) cases of segmented images, along with the respective cropped *ADD* inputs and labels.

5.2.2 Single-frame classification

Training the adapted UNET for classification (Figure 4.8), initialised with the best weights obtained from the segmentation training and freezing 16 layers of the pre-trained UNET encoder, led to significant gaps between training and validation losses (Figure 5.11). This gap was lower for the *DIF* input type, specially the non-cropped one.

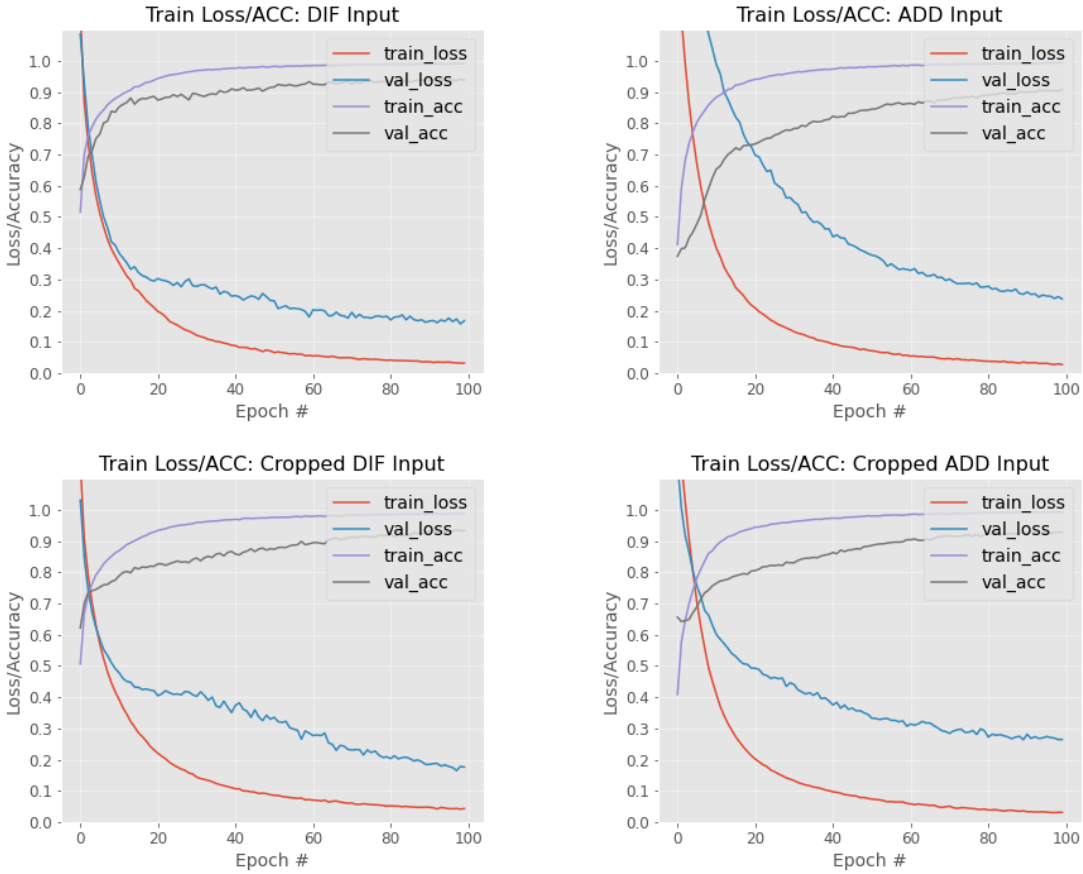


Figure 5.11: Accuracy and loss training curves for the adapted UNET model for classification.

Table 5.10 presents the validation results, as well as the training time for 100 epochs, for every input. The number of epochs was kept constant across all the classification models trained in this dissertation.

Table 5.9: Validation results of the adapted UNET classification model, following the segmentation task, as well as the training time for 100 epochs

Input Type	Crop	ACC (%)	Loss	F1-Score	Precision (%)	Recall (%)	Training Time (h)
DIF	False	**94.09**	**0.16**	**94.14**	**94.29**	**93.92**	4.48
DIF	True	93.47	0.17	93.36	93.70	93.02	4.53
ADD	False	90.82	0.24	91.08	92.07	90.23	4.49
ADD	True	92.79	0.27	92.69	93.08	92.34	**4.43**

In the validation, the *DIF* input attained the best results, with f1-score values of 94.14% and 93.36% for non-cropped and cropped, respectively.

Details on the evaluation of this model in the test set can be seen in Table 5.10 and Figure 5.12.

Table 5.10: Percentage of wrongly classified frames in the test set, by the adapted UNET classification model, following the segmentation task

Input Type	Crop	Wrong predictions (%)
DIF	False	4.71
DIF	True	7.94
ADD	False	**3.19**
ADD	True	3.98

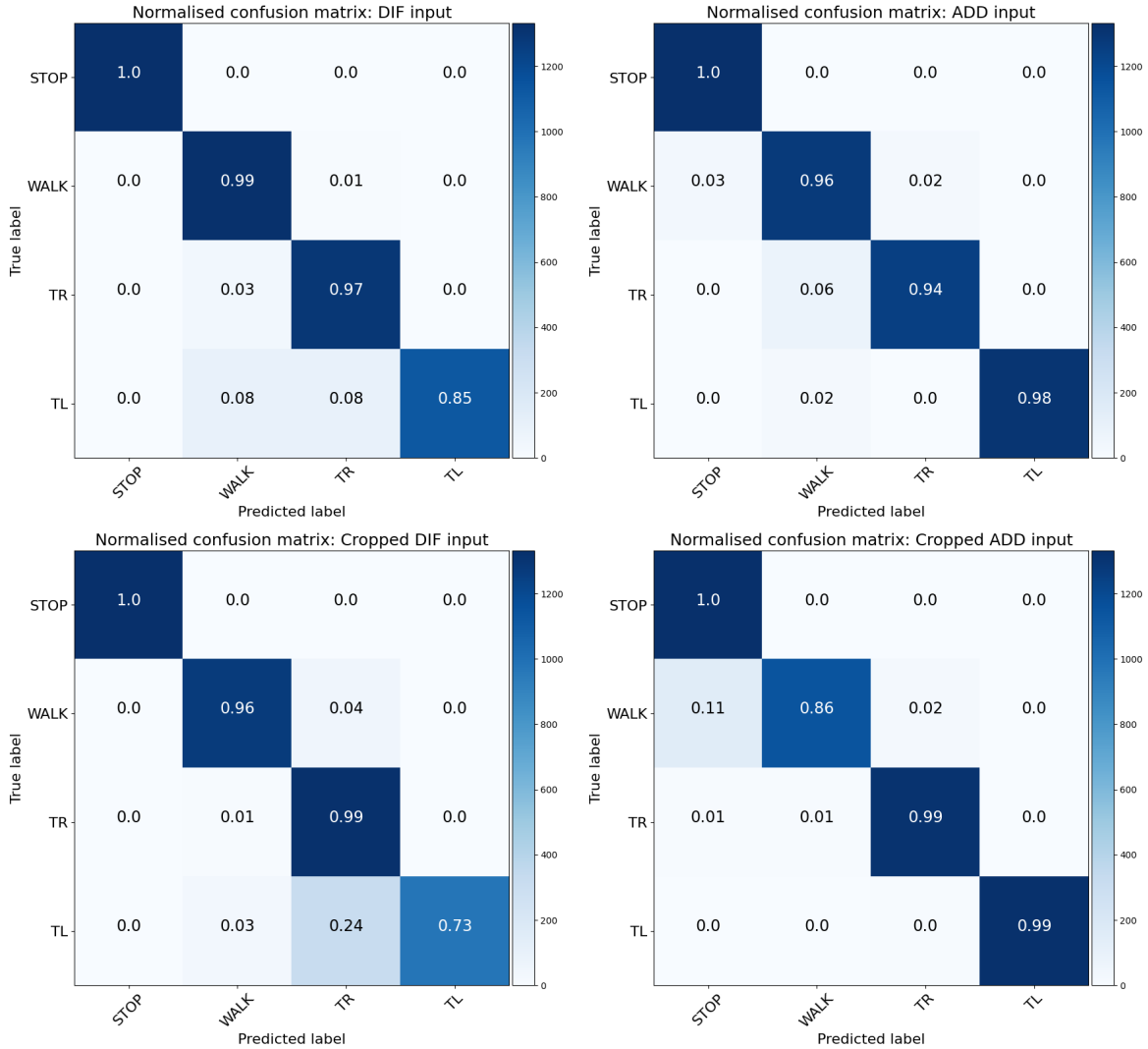


Figure 5.12: Confusion matrices for the adapted UNET classification model, following the segmentation task, over the four types of input.

Contrarily to validation results, Table 5.10 shows a lower percentage of error for the *ADD* input, when classifying the test images (3.19% and 3.98% for non-cropped and cropped, respectively). Moreover, when observing Figure 5.12, it is perceptible that cropped *ADD* input achieved higher or at least equal TP rates

than the other inputs, except for the WALK class (0.83), which was often confused with the STOP and TR classes. It is important to remember that the test dataset comprises more data and subjects than the validation one.

The quantitative evaluation of this model's focus is presented in Table 5.11, where, once again, the best results were achieved with the cropped *ADD* input (Mean Dice values higher than 27.89%).

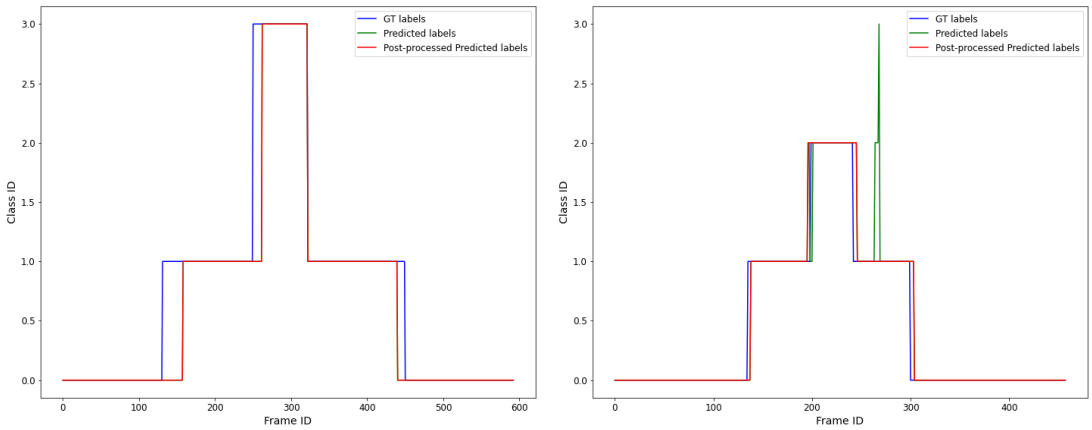
Table 5.11: Quantitative evaluation results of the validation and test grad-CAMs, when predicting with the adapted UNET model for classification

Input		Validation		Test	
Type	**Crop**	**Mean Dice (%)**	**Mean IoU (%)**	**Mean Dice (%)**	**Mean IoU (%)**
DIF	False	19.48	11.01	16.94	9.43
DIF	True	26.99	15.91	26.21	15.56
ADD	False	21.21	12.08	21.68	12.57
ADD	True	**28.17**	**16.75**	**27.89**	**16.61**

5.3 Real-Time Simulation

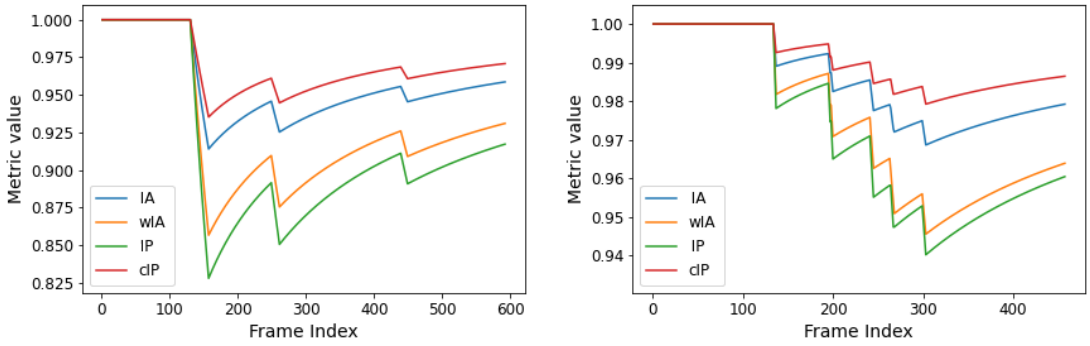
Comparing and analysing the exhibited results on the previous sections, one can infer that the best classification performance, as well as the most relevant and human-centred focus, was achieved by the ResNet-50 model with a channel-wise attention mechanism, fed with cropped *ADD* images (Section 5.1.2). Therefore, this was the approach tested in real-time simulations, along with the post-processing technique depicted in Section 4.4.1. Two examples are presented in this Section, corresponding to trials from different test subjects, with different circuits and velocities: **trial A)** subject 11 performs a turn left at 0.5m/s (lowest gait speed); **trial B)** subject 15 performs a turn right at 1m/s (fastest gait speed). These two trials also intend to represent two different levels of noise/uncertainty in the predictions.

Figures 5.13 allows the comparison, at each instant, between the online predictions (with and without post-processing) and the GT classes, while Figure 5.14 shows the temporal evolution of the online metrics described in Section 4.5.3. Table 5.12 shows the average values for these metrics, for each trial. Note that these metrics were computed considering the final predicted classes (after post-processing).



(a) Trial A. **(b)** Trial B.

Figure 5.13: Plot of the GT, predicted and post-processed predicted labels (Class IDs: 0=STOP, 1=WALK, 2=TR, 3=TL).



(a) Trial A. **(b)** Trial B.

Figure 5.14: Plot of the values of the online metrics described in Section 4.5.3

.

Table 5.12: Average of the online metrics, overall trial

Trial	IA (%)	wIA (%)	IP (%)	cIP (%)
A	95.86	93.10	91.72	97.08
B	**97.92**	**96.39**	**96.04**	**98.65**

Table 5.13 shows, respecting the trials' temporal order, the delays between the post-processed label and the GT one, for each action transition. Negative values represent classes predicted earlier than their actual start.

Table 5.13: Delays of the final predicted labels in relation to the respective GT labels, computed for each transition of the circuit

	Time delay (s)			
Trial	**Walk**	**Turn**	**Walk**	**Stop**
A	1.80	0.80	0.00	-0.67
B	0.20	-0.20	0.27	0.27

5.3.1 Grad-CAMs visualisation

To better understand the model's decisions and if these are based in human-centred features, the grad-CAMs were computed for the ResNet-50 with attention. These can be visualised, in this section, for the transitions in each of the trials in Figure 5.13.

Figure 5.15 presents the grad-CAMs visualisation for **trial A**, where the model's predicted labels correspond exactly to the post-processed ones. For the beginning of each class, the visualisation starts at the first frame of that action (for delayed predictions) or at the first correct prediction (for early predictions, as it is the case of the STOP class, in this trial) and ends at the first right prediction or first GT frame, respectively. Note that these are not necessarily consecutive frames on the dataset, that depends on the class's delay registered in Table 5.13. Nonetheless, they serve as a good representation of the focus evolution between the GT and its respective correct prediction (or vice-versa) and, for the delayed predicted labels, it always include 2 immediately preceding frames.

The same applies to Figure 5.16, representing **trial B**. However, this trial presents more on-off noise in the model's outcomes, specially in the transition from walk to turn (Figure 5.13b). Hence, these predictions do not always correspond to the post-processed labels, so the latter was the one used to mark the start or end frame of these visualisations. Moreover, in the two first presented classes, a frame immediately after the correct post-processed prediction/GT label was added for purposes of focus evolution assessment.

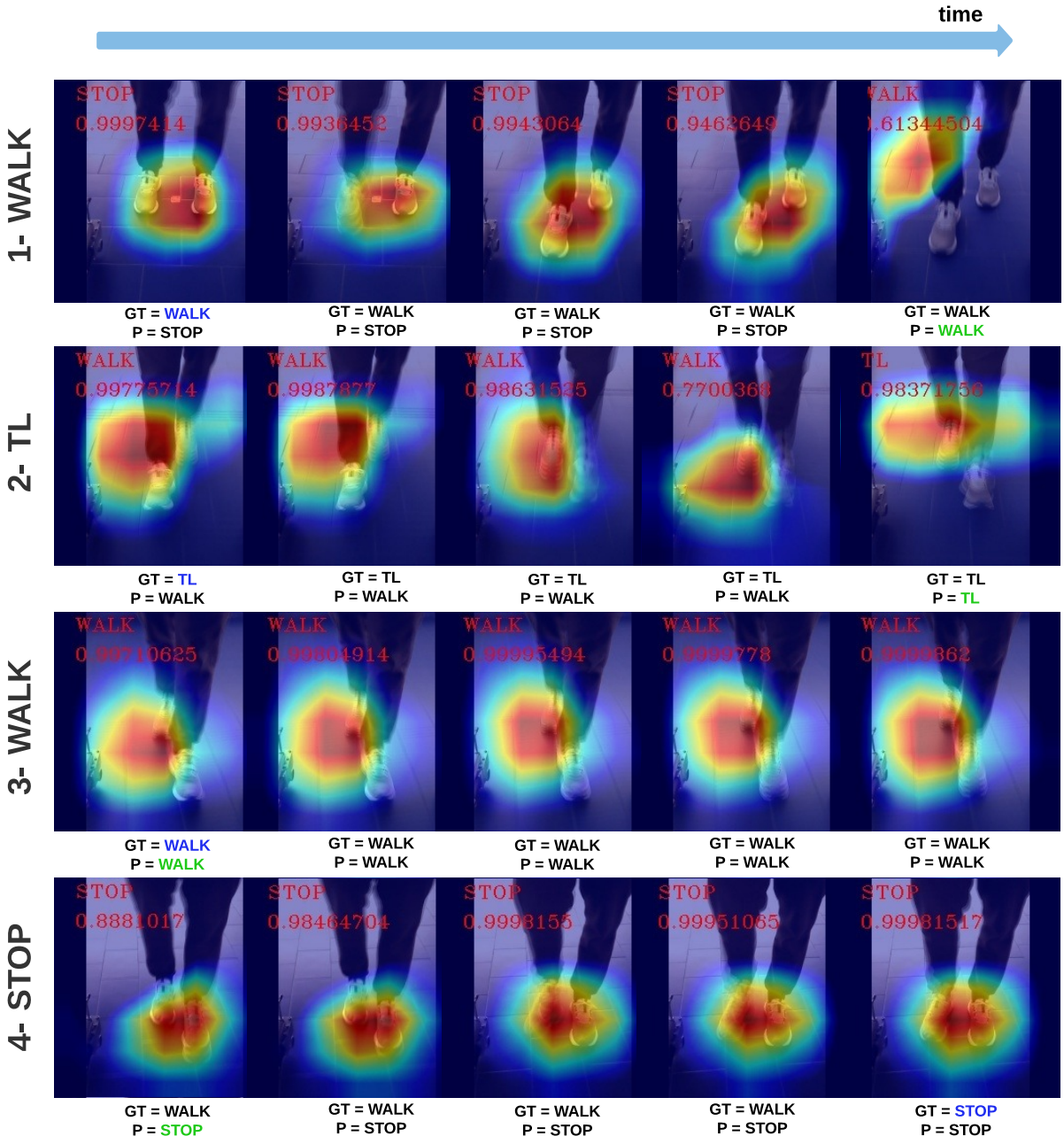


Figure 5.15: Grad-CAMs visualisation, temporally ordered, for each one of the transitions in the slow trial (**trial A**). The green and blue labels correspond to the first prediction and GT label, respectively, of the action that is beginning (P=predicted class).

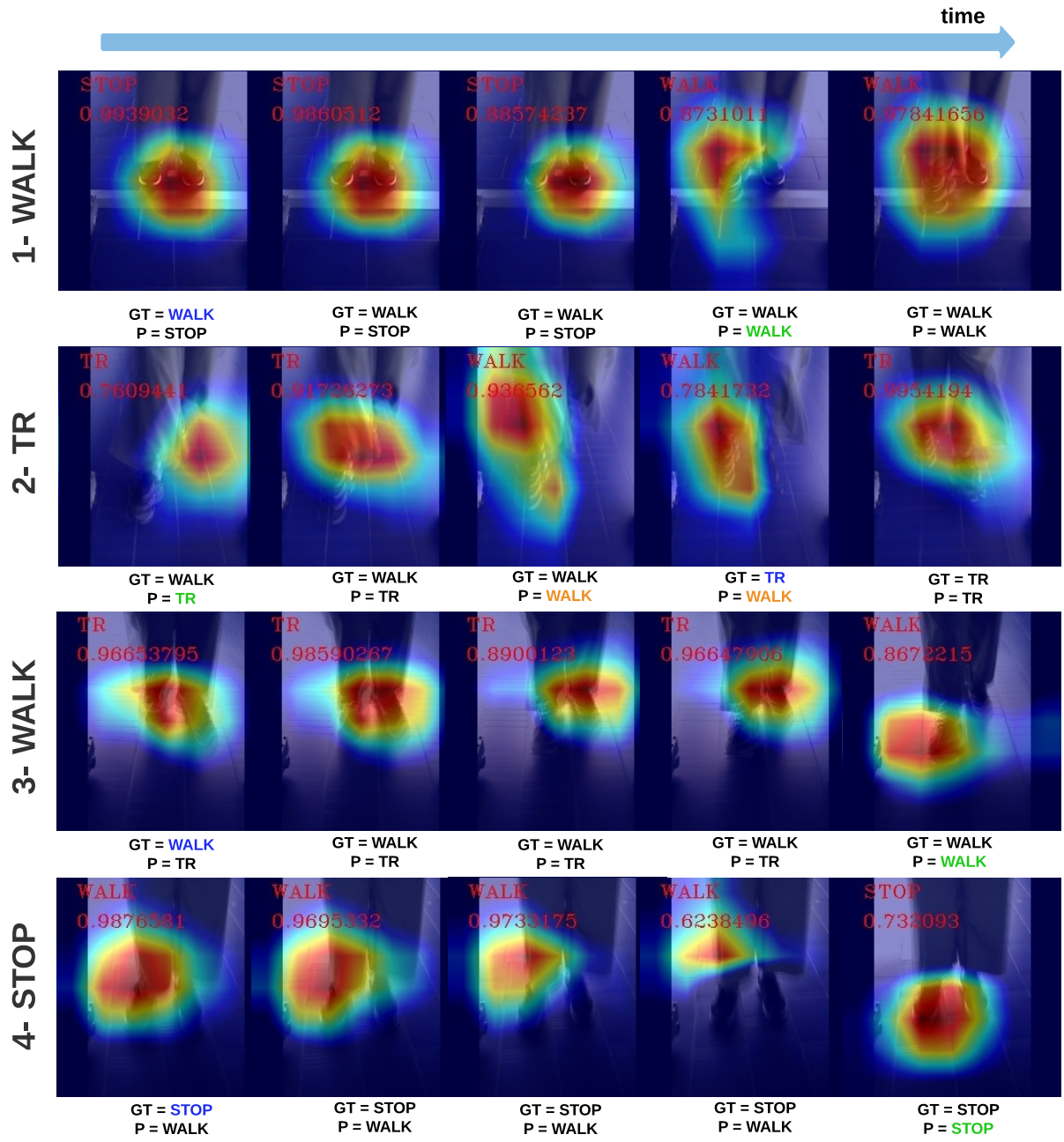


Figure 5.16: Grad-CAMs visualisation, temporally ordered, for each one of the transitions in the fast trial (**trial B**). The green and blue labels correspond to the first prediction and GT label, respectively, of the action that is beginning (P=predicted class). The orange ones correspond to the perturbations in the model's predictions that don't correspond to the post-processed predicted class.

5.3.2 Inference time

The total inference time, using the chosen ResNet-50 with an attention mechanism and cropped *ADD* input, was computed and averaged on a Google Colab instance (Section 4.5.2). The results are exhibited in Table 5.14, where the preprocessing pipeline corresponds to the one in Figure 3.9 (Section 3.4.2),

excluding the augmentation step, and the predicting time includes the model inference, followed by the post-processing. It is noteworthy that the input computation corresponds to the addition of all the 4 frames, being this procedure already included in the preprocessing pipeline.

Table 5.14: Average time to perform each task involved in inference, as well as the total average

	ADD Computation	**Preprocessing Pipeline**	**Predicting**	**Total**
Average time per frame (ms)	5	35	26	61

Chapter 6

Discussion

Throughout this dissertation, several DL frameworks and architectures were proposed for early recognising and detecting human motions from RGB camera streams. These were evaluated for the accuracy of their predictions and the ability to focus the most relevant features of human legs and feet. Along the Chapter 5, a ResNet-50 model with an attention mechanism was highlighted, due to its promising results and thus its performance in real-time simulations, along with the inherent capacity of anticipating human motions, was tested.

In the following sections, the obtained results will be discussed, as well as some limitations and insights for possible improvements of the solutions presented in the previous chapters. The discussion was divided into 5 sections, with the following addressed topics: **i)** data and acquired dataset (Section 6.1); **ii)** inputs attributes and their influence on the models' performances (Section 6.2); **iii)** analysis and comparison of models/approaches and the obtained results (Section 6.3); **iv)** adaptability of the best approach in real-time scenarios (Section 6.4); **v)** general study considerations, as well as suggestions for future research (Section 6.5).

6.1 Dataset

The acquisition method has some limitations, as described in Section 3.2. Firstly, the labels are not completely accurate, due to the difficulty of marking consecutive walking events in real time, as well as the bias introduced by the subject responsible for this procedure. For this reason, the model was not trained with transitions, avoiding the risk of having mislabelled samples deceiving it. Secondly, although the use of the smart walker in active mode avoids the unnatural gait pattern provoked by pulling the passive device,

it also implies that a part of the transition step can be recorded while the walker is already performing the next action. This is most critical in turning events and higher gait speeds, where, despite the user's interaction with the robot and the marked circuit, the visible changes that indicate the beginning of a turn (variation in legs or feet orientations and positions) may be (partially) seen by the walker, when this one is already turning. Hence, the resultant camera's point-of-view and distance to the subject may be different from the ones expected to be found in the real-time application, where the smart walker is still moving straight until it is able to detect the turn. This also adds some background motion to the transition events, which could influence the model, misleading it to look to this background movement.

Therefore, a more accurate labelling strategy is needed, capable of consistently marking the actions in the correspondent toe-off and heal-strike events. A possible solution would be the use of Force Sensitive Resistor (FSR) sensors in the user's shoes, to register the foot contacts without failures (as occurred with the Xsens). This would allow not only the inclusion of transitions in the offline training, but also the study of the step(s) before the transitions to assess if there's any information that would help anticipate the action.

The recording environment, although realistic, provided very few and identical scenarios, always presenting a floor with stripes, which enhances the presence of background motion in the inputs and the chances of this influencing the model's focus. These factors led the model to focus more on background, where relevant features were found. Furthermore, the uncontrolled light conditions lead to corruptions in some depth images and thus in the GT masks used for segmentation and grad-CAMs evaluation, decreasing the dataset size for these tasks. This reduction is also enlarged by the impossibility of setting a constant threshold value to only exclude the corrupted masks (Section 3.4.3), since clothes and heights vary across the participants, leading to the removal of perfectly fine masks. Hence, a more controlled acquisition environment, with a non-marked floor or too bright conditions, could attenuate these constraints, as well as the use of typical and tighter clothes for rehabilitation during the acquisition.

Besides the lower size of the segmentation/grad-CAMs evaluation dataset, the masks may not be perfectly computed, due to the depth corruption issue mentioned before. Even with the mask correction algorithm (Figure 3.12), the user's legs and feet outlines may not be completely correct, which can slightly affect the segmentation and grad-CAMs evaluation outcomes.

It is also important to note that turning with the SW, specially for impaired subjects, is a slower process, segmented in more and smaller steps. This increases the difficulty of turn detection, specially when using only front-view vision-based inputs. Moreover, the steps or starting foot were not controlled, which increases turns variability, but can include more subtle changes, for example, when the turn is started with the "inner foot" instead of the "outer" one.

Collecting data with the subjects always grabbing the device's handles decreased the movement and pose variability available for analysis, but it was consistent with its use in later stage rehabilitation sessions, for balance and stability purposes.

6.2 RGB Input forms

Analysing the obtained results can help to better understand the properties and influence that the different inputs inflict on the models' focus and performances. In Table 5.3, one can see that cropping the images helps to direct the focus to the ROI, as it excludes a significant portion of background. A greater improvement was registered when cropping the *ADD* input, which confirms that this kind of input includes more information about the background motion. This is also confirmed, by the fact that the highest improvement rate in focus was achieved with **approach 2**, a ResNet-50 model with an attention mechanism, with the non-cropped *ADD* input (Table 5.6).

In general, the cropped *ADD* input was the one which presented higher similarities between grad-CAMs and GT masks, across models and evaluation datasets, raising the belief that *ADD* images also encode more human body motion information.

In the presence of a model with attention, which aims to better learn the different discriminative powers among features, both cropped inputs lead to higher and very similar grad-CAMs evaluation results (Table 5.6), showing that also *DIF* has a similar potential of guiding the model's focus, when background motions are duly mitigated and the algorithm properly models the different features importance.

The ResNet-50 model (with and without attention) verified a less relevant focus, when using non-cropped *ADD* frames instead of an input with less encoded human motion (*DIF*). Also, the only time this model attained a worse grad-CAMs evaluation than VGG16, was when their inputs were non-cropped *ADD*s from the test set (Table 5.3), despite the lower rate of incorrect predictions (Table 5.2). Therefore, better classification performances achieved by this input form are not so reliable.

From these comparisons, one can infer three aspects: **i)** background motion may contain more evident features, easier to extract, that can help the offline classification task. However, this should not be the main focus of the model, as these features are not reliable for transitions, real-time applications or even generalisations to other datasets, where the background is static; **ii)** dealing with inputs that encode more overall motion information (non-cropped *ADD*), can deviate the model's attention, which was the case for the ResNet-50 model (with and without attention). So, a carefully performance evaluation is needed, as better classification results can be associated with non-ideal feature extractions and overfitting to the

background; and **iii)** despite not being a commonly used approach, cropping the inputs was a feasible way to enhance the model's focus and thus increased the reliability of the respective classification results. Nevertheless, the higher registered improvement was not higher than 10%, although this can be related to the floor characteristics discussed in Section 6.1.

One could argue that using depth data to preprocess the RGB frames, removing the background and, therefore, isolating the user, should have also been considered and experimented. However, besides the extra computation that this would require, depth images can be corrupted in non-controlled environments, leading to corrupted masks (case in point for Figure 3.15). Introducing these failure cases would decrease the model's robustness and/or the size of the dataset.

Performing a comparison with the literature, state-of-the-art articles for action recognition/forecasting (Table 2.3) normally use multiple RGB frames to provide the sense of temporal motion, while the proposed inputs were able to provide (a significant part of) this information in one single frame. With a larger background variability in the dataset, while carefully avoiding pavement marks during the acquisitions, these tailored RGB inputs, with only one frame each, could become a more reliable method to induce action-aware feature extraction.

6.2.1 Grad-CAMs evaluation algorithm

The developed grad-CAMs evaluation algorithm establishes a good comparison term between models evaluated in the same dataset, as it used the same GT masks and inputs. It then computes the similarity metrics between these binary masks and the respective grad-CAMs' heatmaps, followed by the mean computation overall frames. Nonetheless, there is an aspect about this method that can decrease the overall metrics: GT masks present the highest score (1) for all the human area, but this region is never equally important to the motion decoding. For example, feet and knees may present more orientation and position variations which indicate the step's direction, so it would be correct for the model to give higher focus to these particular regions. For this reason, comparing heatmaps to these masks may be correctly penalising FP, but it is also counting with smaller differences between TP that do not hold the highest score in the heatmap, while this could not be entirely wrong. So the model can be focusing on human pixels and still being a little penalised for it, in the final evaluation. Or it can be rightly and exclusively extracting features from the feet, while still being penalised by not focusing on the rest of the legs, wrongly considering these heatmap's pixels as FN. The masks are already computed in the tightest ROI possible to attenuate this effect, but it does not completely solve it.

A possible solution to overcome this limitation would be to convert the heatmap into a binary matrix,

so the TP would never be penalised. However, this would still not be evaluating if the model is focusing on the human areas with higher motion, while also attributing no penalisation to possible human motion relevant regions with a heatmap score significantly lower than 1. It would also imply that the model should focus on the whole foreground, which is not true. Therefore, a more reliable solution would be to change the GT masks pixel values, according to the input images. Thus, as the higher pixel intensities in the input correspond to motions with larger amplitudes, while the lower correspond to more static areas, this information could be used to scale the scores equal to 1 in the GT masks, creating a sort of *human motion masks*. An even more accurate form of information to scale these masks according to the body's amplitude of motion, would be the human poses, from the Xsens data, for example. Nevertheless, this last option would unduly increase the computational expense and complexity of this algorithm.

Nonetheless, the most important is to not focus on the background, as perhaps it would be safer to leave the foreground prioritisation criteria for the model to decide. So one could just evaluate the percentage of FP. As part of future research, these two last suggestions could be experimented at the same time to evaluate the model's focus, while assessing the changes in the reliability of this evaluation method.

6.3 Models performances

Baseline CNN models

Based on the results presented in Section 5.1.1, it is possible to infer that ResNet-50 performed better than VGG16, achieving f1-score values between 94.34% and 98.27%. Hence, this problem benefited from residual features, skip connections and deeper networks duly initialised or pre-trained. Nevertheless, the difference between both performances was not that high (lower than 1.47% in f1-score), meaning that the task of early recognising human walking motions from the WALKit SW dataset may also be approached by less deep models. Not only this would reduce the number of parameters to train, but it could perhaps extract more general features from images, improving the generalisation power, when increasing the dataset's variability [29].

ResNet-50 also presented better focus (maximum Mean Dice of 32.13%), when compared to the VGG16. However, there was an exception: when dealing with a bigger evaluation set (test set) and with non-cropped *ADD* inputs, this model extracted lesser human-centred features, achieving a Mean Dice of 16.93%, while VGG16 attained 20.99% (Table 5.3). As discussed in Section 6.2, the *ADD* corresponds to the input form that encodes more motion information, from both background and foreground. So,

this shows the tendency of the model to rely on background features, when these are more evident and available. It also shows that sometimes better classification rates may be deceiving, as the model can be supporting its decisions on the less relevant information.

Nonetheless, ResNet-50 achieved better classification and focus results, over the majority of the inputs and evaluation sets, and was thus the chosen model to be tested with an attention mechanism (Section 5.1.2).

Channel-wise attention mechanism

The addition of the channel-wise attention mechanism enhanced, not only the classification metrics, improving the f1-score by an average of 2.93%, but also the similarity between grad-CAMs and GT masks, with improvements until 5.21% in Mean Dice, across all inputs. Only in the validation set, the model's focus associated with the cropped *ADD* input was slightly worse than the ones registered with the ResNet-50 baseline model (see Tables 5.3 and 5.6). But, since this is the smallest evaluation dataset and the difference is not significant (0.34% and 0.68% in Mean Dice and Mean IoU, respectively), these could be due to small variations and, thus, were not considered as a relevant fact.

These results proved the importance of dealing and modelling the distinct learning abilities of the different convolutional channels, not only to increase CNN performance, but also to improve the relevance of the features extracted. However, the values presented in Table 5.6 are not that higher than the ones in Table 5.3, specially for the cropped inputs, proving that: **i)** cropping inputs is effective when it comes to decrease focus deviations, achieving improvements up to 9.57% in Mean Dice, for the baseline models; **ii)** this channel-wise attention mechanism, although unequivocally beneficial to the classification task, still does not completely correct its main focus, as the maximum value of Mean Dice was still 22.70% lower than 55%.

Facing these facts, future research could integrate the design of a spatial attention mechanism to tackle this problem, guiding the model to use *intentional regions*, instead of the whole frame. As in [70], also enhancing local features by combining this with the channel-wise mechanism, could lead to better performances and, in this case, more properly focused solutions. The spatial attention maps could even be compared with the suggested *human motion masks* (Section 6.2.1) for focus evaluation or, in a more bold experiment, as part of the model's loss.

Nonetheless, this model stood out as the most promising one, while still preserving low complexity traits, when compared to the literature on this topic (DL solutions for human action recognition or future action prediction, see Table 2.3). While the state-of-the-art models commonly resort to LSTM layers or

3D-CNN, here the classification is performed with only one 2D-CNN model. When facing the need to further reduce the model's complexity, one could also add this attention mechanism to the VGG16 model, without excessively compromising the final performance. Although the obtained results can be considered as being in agreement with those presented in the literature, this is not a fair quantitative comparison, since different evaluation protocols, including datasets, were used. Therefore, benchmark studies should be performed in order to allow a more reliable comparison with these state-of-the-art approaches.

Segmentation-Classification approach

Looking at the segmentation training curves (Figure 5.6), one can see that, as the epochs advance, there is a slightly growing tendency for overfitting. Although apparently small, this can propagate to the following pre-trained classification model and induce bad generalisation abilities or even worse cases of overfitting. That's why the segmentation training was shorten to 30 epochs and the weights were chosen considering the minimal validation loss.

As for the examples of segmented images (Figures 5.7, 5.8, 5.9 and 5.10), it is visible that the worst case always corresponds to the same subject who was wearing very large pants, when compared to the best segmented cases. This confirms the disadvantages of not controlling the wardrobe, during data acquisitions.

Despite the training reduction, the adapted UNET still revealed problems of bad generalisation (Figure 5.11). The severest cases of unrepresentative training datasets and consequent generalisation issues were verified by the *ADD* input type and these cases are associated with lower validation performances, namely 92.69% (cropped) and 91.08% (non-cropped form) of f1-score (Table 5.9). Nevertheless, when observing the test results (Table 5.10 and Figure 5.12), these inputs attained and even surpassed the *DIF* ones. This points out to the need of implementing cross-validation in future assessments, as the validation set may not be representative enough.

When connecting the segmentation (Tables 5.7 and 5.8) and classification results, the *ADD* input type appears as the easiest to segment, showing the highest values of Mean IoU and Mean Dice, for both validation (86.39% and 96.23%, with the non-cropped form) and test sets (92.15% and 95.88%, with the cropped form). However, it also appears to have led to worst cases of weak generalisation (Figure 5.11). Contrarily, the non-cropped *DIF* images achieved the worst segmentation Mean Dice, across the two evaluation datasets (95.17% and 93.74% for validation and test, respectively), which lead to a smaller gap between loss curves (Figure 5.11). So, it seems to exist an inverse relation between segmentation power and the classification's generalisation ability. This may mean that this cascade approach is leading

the model to focus on input traits that are not representative of the whole dataset, following the overfitting problems during segmentation.

Nevertheless and as mentioned before, for one to be completely sure about this last statement, cross-validation should be performed, in order to avoid low representative validation datasets. Still, the focus on particular traits may be associated with the fact that, despite the final aim of human motion decoding, the GT masks used are leading to the segmentation of the whole body, including large clothes and static human areas. Therefore, using the mentioned *human motion masks* as labels (see Section 6.2.1 for details) would decrease the chances of overfitting, while pursuing the differential segmentation of the human body, according to its motion. This could enhance the weights used to pre-train the classification model. Other options to help overcoming the overfitting problem consist on experimenting other simpler segmentation models or even include spatial data augmentation. Moreover, the number of frozen layers should also be studied and tuned.

Once again, a fair comparison with state-of-the-art classification models cannot be established. It can only be inferred that this adapted UNET model has a few less parameters than the Y-Net [84], while the latter presents residual convolution blocks and is jointly trained for both tasks. Hence, as pointed out before for segmentation, overall less complex architectures could be tested, in an attempt to reduce overfitting and increase the reliability of the classification results.

In agreement with the training curves (Figure 5.11), the classification metrics were worse than the ones achieved by previous evaluated models, as the maximum f1-score was of 94.14% (Table 5.9), which is lower than the minimum registered for the previous models (94.34%, for the baseline ResNet-50, shown in Table 5.1). Nevertheless, these metrics were still above 90%. As for the grad-CAMs evaluation (Table 5.11), this approach was not able to significantly increase their similarity with the human masks, as its results were not so different from the ones obtained by VGG16 (Table 5.3). Moreover, it attained even worse results than ResNet-50 mode, without (Table 5.3) and with attention (Table 5.6). The only exception was the non-cropped *ADD* input in the baseline ResNet-50 model, which lead to worse focuses than in VGG16 and in this adapted UNET. This may raise the possibility of these simpler classification models being able to ignore a little more the excess of background motion information (for example, the small stripes movements). Nonetheless, it does not provide evidences that this segmentation is helping to guide the classification model's focus, since this difference between results seems to be mainly influenced by the input properties, for their lower values and resemblance to the VGG16 ones. Therefore, it is safe to say that the segmentation-classification approach was not the most effective approach when attaining its main goal: improving the extraction of human-centred features to distinguish between actions.

6.4 Real-Time Simulation

The ResNet-50 model with a channel wise attention mechanism was the best model in both aspects: classification rates and focus relevance, specially when fed with the cropped *ADD* input. Testing this approach in real-time simulations lead to good performances, with the model being capable of identifying the different consecutive walking events. In some trials, such as **trial B** (Figure 5.13b), the model's outputs revealed its uncertainty in the form of on-off noise. This was shown to be easily corrected by post-processing. Besides not adding significant computations, the implemented post-processing technique (Section 4.4.1) matched the action beginnings marked by the predicted labels, not increasing the delay inherent to the model's decision making process, which is the usual disadvantage of post-processing implementation.

Every trial starts with online metrics of 100%, since the model can easily detect the STOP class. Nonetheless, as the trial proceeds, these values drop, specially in transition frames, due to the delays registered between the predictions and the GT classes. These metrics recover during the action, achieving averages higher than 90% (see Figure 5.14 and Table 5.12). Despite not presenting any noise, due to its higher delays, **trial A** achieved lower metric values (average IA and IP of 95.86% and 91.72%, respectively), when compared to the post-processed predicted labels in **trial B** (average IA of 97.92% and IP 96.04%).

The beginning of each class is normally predicted later than the actual GT, but the registered delays were not that high ([0.0, 1.8]s). For the trial performed at 1m/s, the delays are at least 0.37s lower than the average step time for this gait speed (0.64s) [1]. Considering the inference time of 61ms (Table 5.14), the complete procedure to early detect an action, since the raw input until the first correct post-processed prediction, ends before that class's initial step is complete. However, this should be tested on WALKit SW, since the inference time on its computer can be higher, given its computational resources (Section 4.5.2).

The delays were higher for slower gait speeds (0.5m/s), as one can see in Table 5.13. Turning with the SW is already more subtle, as the described curve is wider. For lower velocities, the variations are even less evident, as the turn is segmented in more smaller steps. Hence, it was expected that the delays would increase with the decrease of gait speed, specially since this model was not trained with transitions (see Section 3.3.2). The average step time for walking at 0.7m/s with the WALKit SW corresponds to 0.77s (approximately), so the time lags were lower than the expected time step for 0.5m/s, except for the first WALK action (delay of 1.8s). For the TL action, the delay of 0.80s is at least equal to this time step.

Overall, the results prove that the chosen approach is suitable for early action recognition, achieving

[1] Determined in laboratory experiments

average online metrics between 91.72% and 98.65%. However, this still needs improvements before being applied for early action detection, as it can be seen by the model's performances at the transition inputs. Nevertheless, the time lags were not so critical, ranging from -0.67 to 1.80s, so perhaps with a proper training procedure that includes transitions in the dataset, this performance could be enhanced.

6.4.1 Grad-CAMs visualisation

The displayed grad-CAMs showed that the model focus is not too deviated from the human region, but it still considers some background information, specially the visible motion of the floor stripes.

For example, in Figure 5.16 (last row), the stop detection was delayed, as the walker kept moving after the subject stopped. So the model must have considered the stripes and the large clothes motions, instead of the human's steadier positions. As the device decelerates, this background motion became less evident and the model started to focus on the user's feet. In the slow trial, this deceleration phase is shorter and slower, so the background motion stopped appearing in the RGB inputs before the human movement did, allowing the model to better perceive the feet becoming more steady and closer to each other (last row of Figure 5.15). This situation is similar to the beginning of the first walking event at low gait speed (first row), where the walker starts to slowly accelerate, so the background appears static, and the feet move slowly and closer to each other, leading to a confusion between STOP and WALK classes.

The turning event was anticipated in **trial B**, which seemed like a good achievement of MI decoding. Nonetheless, looking at the respective grad-CAMs (second row, in Figure 5.16), one can see that this class was first predicted based on the vertical misalignment between the floor stripes. This helps to visualise and understand the confusion and model's uncertainty between these two classes (WALK and TR/TL). Likewise the **trial A** (second row, Figure 5.15), the model's turn predicted label only stabilised closer to the end of this event's first step, which is not ideal for a SW's control purposes.

Another critical prediction, when controlling a SW, will be the walking straight event after a turn. If this label is delayed, specially when moving at faster velocities, the walker will keep turning, leading to undesired and non-controlled trajectories. In the slow trial, this class was early detected in time (relatively to the GT label), but in the faster one, where this issue is more critical, it was only detected at the end of the first step of the walking action (third rows, Figures 5.15 and 5.16, respectively). However, in the latter case, the right foot appears to be pointing right, which can be deceiving even for humans. People with feet rotational disorders can thus mislead the model.

Although the chosen trials are representative of the test set, corresponding to different conditions and extreme cases, one could perform more simulations with other trials and subjects to better evaluate the

general focus and time delays inherent to the model's performance. These visualisations showed that the model's focus still needs to be improved in transitions, in order to be implemented as a real-time control mechanism. In that situation, the subject will be giving the motion intention before the walker starts to perform the respective action, so the background motion, specially the stripes movement, will not be present at that time, since this is a consequence of the SW's movement.

6.5 General Considerations and Future Research Suggestions

Instead of using only train/test split, cross-validation should be implemented during the training phase, to assure more certainty about the model's performance and better sustain the choice of the most suitable approach, using the test set only for final evaluation purposes.

Observing the confusion matrices and the real-time simulations plots, one can perceive than the STOP class is perfectly distinguished from the other ones. Walking straight is though confused with turn events, as they both present background motion and similar leg and feet positions, since the turnings present more subtle human body variations, when executed with a walker and visualised only in a front-view perspective. Thus, this algorithm would be perfectly suited for a two-class application, distinguishing between STOP and WALK.

Inspired by [14], this dissertation also attempted to explore action-aware features, but instead of deploying more complex models (two CNN branches and 2-stage LSTM), here different forms of RGB input were designed, as well as approaches to orient the model's focus to the human body region, while the grad-CAMs were only used for focus evaluation purposes. All of these solutions were developed, aiming less complex architectures, lower computational times and the ability to be implemented in online applications. It is common to find in the literature special designed losses to enhance the early action detection or action anticipation tasks (Section 2.3.3). Although the scope of this dissertation was directed to tailored inputs and mechanisms to enhance a human-centred feature extraction, tailored losses could also be tested in the future, specially if these methods still leave room for time delays reduction and thus improvement of the model's early detection ability.

Although encoding relevant motion information, the *DIF* and *ADD* inputs share a common disadvantage with the optical flow (OF): in realistic videos, where there is usually camera motion, these forms of input can also encode background movement and may not concentrate on the human action. This may help identifying ongoing actions, but deceives the model with non-relevant features, when predicting transitions or future actions.

Despite all the tests and designed approaches, a mainly action-aware feature extraction was still not achieved. Therefore, to improve the features relevance, self-supervised learning techniques (for example, the contrastive embedding method [87][88]) could be explored, in order to learn good representations of the input that could be then used in a supervised learning task. Nevertheless, these architectures are more complex, requiring large amounts of data (specially, a large number of negative samples) and more computational resources. Moreover, self-supervised learning is still being explored on the continuous world (for images, videos, among others), so this would require a literature review on this topic, along with the collection of more data.

Chapter 7

Conclusion

In this work, a novel vision-based DL solution was developed to tackle two problems at the same time: human motion decoding and early action detection. It was able to understand the walking event tacking place, by only seeing small windows (length of 4 frames, with stride of 2) of lower body RGB streams recorded by the WALKit SW. The solution is adaptable to real-time scenarios and thus capable of, with further improvements, integrate a human-in-the-loop control strategy to drive the SW. The developed work was based and inspired in an extensive literature review on MI decoding algorithms in SWs, as well as action recognition and forecasting with vision-based DL models. This helped to understand the so far implemented techniques and inherent limitations, providing insights about the promising approaches that could be implemented and the areas more prompt to innovation and enhancement.

Custom modules for acquisition, including protocols, an *automatic trajectory mode* to automatically drive the walker and a real-time automatic labelling procedure, were devised to collect data in realistic scenarios and circuits, without modifying people's natural gait and while promoting their interaction with the front-following robot and the defined trajectories. A dataset of 15 healthy adults was then acquired, with 24 trials per subject, each one containing three of the target classes (STOP, WALK, TR/TL). From this, a balanced dataset of frames was created to finally train, evaluate and compare the proposed approaches, with a total of 28800 RGB images.

A novel method to quantitatively evaluate the model's focus was designed, where the model's grad-CAMs, for each input, are computed and compared with the respective GT masks. Since this dissertation aims to recognise the user's motion intent from RGB frames, assessment of the model's ability to enhance and use mainly local relevant features, i.e. from human motion, is of high importance.

Multiple approaches, including designed motion-encoding RGB input forms and several model archi-

tectures, were trained and evaluated in the WALKit SW acquired dataset, achieving, in general, good classification performances (with accuracy and f1-score higher than 90%). The best results were obtained by a ResNet-50 model with a channel-wise attention mechanism, fed with cropped *ADD* images, reaching a f1-score and accuracy of 99.61% and over 30% of Mean Dice between the model's grad-CAMs and the GT human masks. This was the input form that lead to the best results, as it is capable of encoding more motion information, while the cropping procedure removes the extra undesired background features.

This final model was tested in real-time simulations, achieving good performances as well, with mostly small delays in the predictions and high online metrics (average wIA $> 93\%$ and average cIP $> 97\%$). This performance was possible through the implementation of a post-processing technique able to reduce the model's uncertainty, while not introducing time lags in the decision process. Therefore, the registered delays are completely inherent to the model's performance and focus, both consequences of the learning procedures and dataset details.

Promising results were obtained to early detect human motions. However, a more accurate labelling procedure and a higher control over the acquisition environment (specially, the absence of floor marks) is essential to train the model with accurately marked transitions, as well as learning weights that do not overfit to the background motion. This would enable a more reliable evaluation of the model's performance and its focus during transitions, while prompting the extraction of more human-centred features. With these enhancements, the final solution could be further assessed, improved (if necessary) and integrated in the WALKit SW.

The presented work allowed to obtain answers for the addressed Research Question (RQs):

RQ 1: How to acquire data with the SW, without significantly disturbing the subject's gait and with a sufficiently accurate automatic labelling procedure?

An *automatic trajectory mode* to automatically drive the SW was proposed, removing the need for the subject to pull the device and thus promoting more natural gait patterns and motion intentions. This active mode allowed to use the walker's velocity information, as well as the joystick signals to create real-time labels focused on the walker's motion and the user's motion, respectively. The Xsens MTw Awinda system also contributed to these labels correction.

RQ 2: Which inputs can be applied to the DL models that entail a low computational load, while encoding the human motion?

Thanks to the front-following action performed by the WALKit SW, the camera moves along with the subject who is grabbing the device's handles, assuming a stable centred region in the camera's field of view. This allowed the creation of perceptible RGB single-frame inputs, encoding the motion information

along a window of frames. By adding and subtracting the present and past images, one can see the evolution of feet and legs positions and orientations in only one frame, discarding the use of RNN. Moreover, results suggest that the *ADD* input encodes more motion information.

RQ 3: How can one improve the model's focus, leading it to mainly extract relevant features from the input's human body region?

Cropping most part of the background surrounding the image's ROI proved to have a major impact on the model's focus, directing it to the human region of motion. Adding an attention mechanism to the model proved effective as well, so the combination of both lead to the extraction of more human-centred features. This can be assessed during the evaluation process, through grad-CAMs computation and their comparison with the respective GT human masks. Nevertheless, the maximum Mean Dice between grad-CAMs and GT masks was only 32.30%, meaning that the designed approaches were still not enough to raise the human-centred focus percentage over 50%, at least. Part of these results is assumed to be derived from dataset conditions, such as the constant presence of floor stripes, enhancing the background motion and consequent overfitting, but also from the grad-CAMs evaluation algorithm limitations (see Section 6.2.1).

RQ 4: Which DL framework produces best results on early detecting the human motion considering a small window of the action?

A ResNet-50 model with a channel-wise attention mechanism, fed with cropped *ADD* inputs computed from a sliding window approach of length 4, attained the most promising results (offline accuracy and f1-score higher than 95%). Nevertheless, the model's focus is still being deceived by the background motion derived from the walker's active mode, when acquiring the dataset (Mean Dice lower than 33%). This is critical when early detecting the motion's beginnings and it was also prompted by the lack of reliable action transition frames in the training dataset. To have more certainty on the reliability of these results, the model's focus needs to be further improved, avoiding these background extracted features that compromise its real-time performance, when transitioning between actions.

RQ5: How effective and robust is the proposed DL solution for real-time applications towards a future human-in-the-loop control strategy?

The model's focus still requires enhancements, to increase the solution's reliability and possibly reduce time lags inherent to a correct early detection, promoting a better real-time performance. The established goals were fulfilled for fastest velocities, where online metrics surpassed 95% and time lags were much smaller than the respective step time ($< 0.64s$ at $1\mathrm{m/s}$). This confirms the greater challenge implied by early detecting slower and more subtle motion changes. Contrarily, the model's uncertainty revealed a greater prominence at higher gait speeds, but these perturbations were easily smoothed by the proposed

post-processing technique, without increasing time delays in the correct predictions.

Considering now the used stride of 2 over camera streams recorded at 30Hz, the inference time should not be superior than the inter-frame temporal distance (67ms). This time was averaged as 61ms, using a Google CoLab instance, but conclusions about this performance in the WALKit SW cannot be extracted. One the one hand, the walker's computer has fewer computational resources. On the other, there are changes that can be implemented in real-time to reduce the computed inference time. For example, gradually adding the frames and saving them while they are being recorded, so the *ADD* computation is practically done by the time the present frame is recorded.

7.1 Future Work

Several suggestions for future research and improvements were raised during this work, but the most imminent to achieved the required performance for real-time applications in a SW are the following:

- Improve the labelling procedure accuracy, for example, with the use of force sensors, to allow the inclusion of transitions, without mislabelled samples.
- Enhance the quality of the acquired data, recording in a more controlled environment, without marked floors or too bright conditions, avoiding the model's overfiting to the background, as well as the depth, and consequently, human masks corruption.
- Improve the model's focus evaluation algorithm, at least by analysing the total FP. *Human motion masks* could also be deployed, instead of only binary masks marking the human body region.
- If one is still not able to increase the grad-CAMs evaluation metrics over 55%, experiment other model architectures, such as spatial attention mechanism, self-supervised learning techniques, among others.
- Perform benchmark and ablation studies.
- Determine the real inference time, in the WALKit SW, to verify if, along with the model's delays, an early action detection is enough, even for the most critical transitions, or if an action anticipation should be forcefully implemented.
- When unable to decrease the inference time below 67ms, one could discard the sliding window approach, introducing a time interval between input computations that is compatible with the normal

minimum duration of human motions performed by impaired subjects (as it was implemented in the designed post-processing technique). Another possible idea would be to combine this with a gait analysis model to only perform the action detection in decisive gait events, such as the toe-off [13].

7.1.1 Human-in-the-loop Control Strategy Proposal

Integrating this classification approach in a driving control strategy could be deployed by implementing a Finite State Machine (FSM) that would relate the predicted classes with the wheels' linear velocities, with respect to the overall linear velocity that is normally defined at the beginning of each trial. In the turns predictions, the wheels' velocities would be computed by the system of equations (3.1) (Section 3.2.6). The FSM would then send these as velocity commands to the respective PID controllers, adjusting the reference velocity for each wheel. Figure 7.1 illustrates this proposal, where the Human Motion Decoding Module corresponds to the developed DL solution, including the preprocessing of RGB raw data into the final normalised input, the model and the post-processing technique.

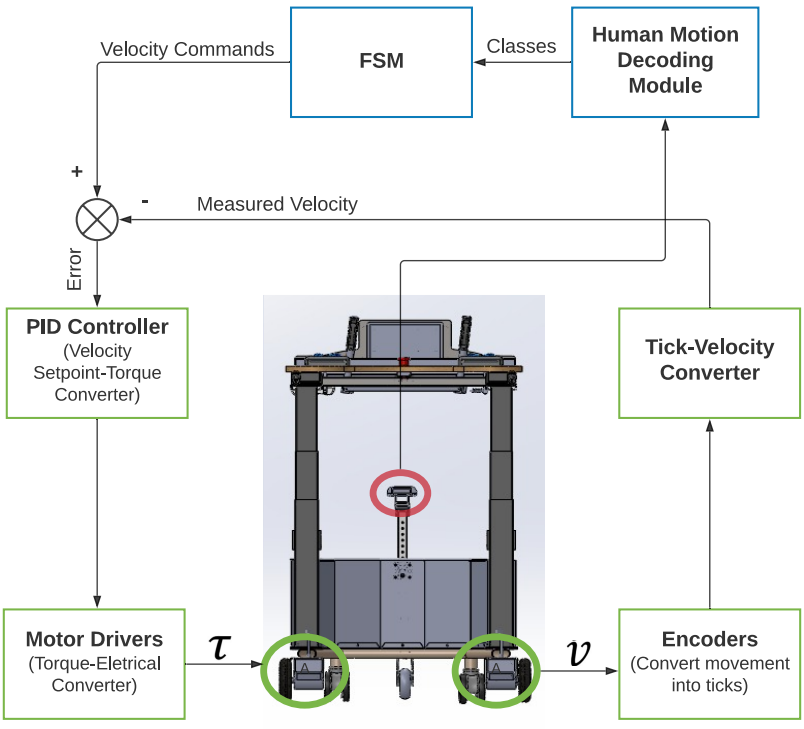


Figure 7.1: Diagram of the proposed human-in-the-loop control strategy, integrating the DL solution for human motion decoding.

Despite being in accordance with the SW functionalities for rehabilitation therapy, a classification solution like this does not allow the automatic control of the walker's velocity, as well. Another idea for a

future deployment could be the development of a regression model, capable of inferring the angular and linear velocities from the camera streams. Inertial data from Xsens MTw Awinda or the walker's velocities recorded when using the *automatic trajectory mode* for data acquisition would serve as labels to predict the gait linear and angular speeds that encode the intended movements. This control strategy would be similar to the one displayed in Figure 7.1, but without requiring a FSM. The velocities predicted from the Human Motion Decoding Module would be used in the system of equations (3.1) to calculate the velocity commands for each PID controller.

References

[1] World Health Organization, *Disability and health*, 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health (visited on 06/10/2021).

[2] T. Mikolajczyk, I. Ciobanu, D. I. Badea, A. Iliescu, S. Pizzamiglio, T. Schauer, T. Seel, P. L. Seiciu, D. L. Turner, and M. Berteanu, "Advanced technology for gait rehabilitation: An overview," *Advances in Mechanical Engineering*, vol. 10, no. 7, pp. 1–19, 2018, ISSN: 16878140. DOI: 10.1177/1687814018783627.

[3] H. Bonney, R. de Silva, P. Glunti, J. Greenfield, and B. Hunt, "Management of the ataxias towards best clinical practice Third edition," no. July, p. 29, 2016.

[4] J. Jonsdottir and M. Ferrarin, "Gait disorders in persons after stroke," *Handbook of Human Motion*, vol. 2-3, pp. 1205–1216, 2018. DOI: 10.1007/978-3-319-14418-4_61.

[5] Y. Celik, S. Stuart, W. L. Woo, and A. Godfrey, "Gait analysis in neurological populations: Progression in the use of wearables," *Medical Engineering and Physics*, vol. 87, pp. 9–29, 2021, ISSN: 18734030. DOI: 10.1016/j.medengphy.2020.11.005. [Online]. Available: https://doi.org/10.1016/j.medengphy.2020.11.005.

[6] Josefina Gutiérrez-Martínez, "Neuroprostheses : Significance in Gait Rehabilitation," *Advanced Technologies for the Rehabilitation of Gait and Balance Disorders, Biosystems & Biorobotics*, no. 19, pp. 427–446, 2018. DOI: https://doi.org/10.1007/978-3-319-72736-3_29.

[7] J. Olesen, A. Gustavsson, M. Svensson, H. U. Wittchen, and B. Jönsson, "The economic cost of brain disorders in Europe," *European Journal of Neurology*, vol. 19, no. 1, pp. 155–162, 2012, ISSN: 14681331. DOI: 10.1111/j.1468-1331.2011.03590.x.

[8] World Health Organization, "World report on disability.," *Disability and rehabilitation*, vol. 33, no. 17-18, p. 1491, 2011, ISSN: 14645165. DOI: 10.3109/09638288.2011.590392.

[9] S. C. Milne, L. A. Corben, N. Georgiou-Karistianis, M. B. Delatycki, and E. M. Yiu, "Rehabilitation for Individuals with Genetic Degenerative Ataxia: A Systematic Review," *Neurorehabilitation and Neural Repair*, vol. 31, no. 7, pp. 609–622, 2017, ISSN: 15526844. DOI: 10.1177/1545968317712469.

[10] E. O. Arogunjo, E. D. Markus, and H. Yskandar, "Development of a Holonomic Robotic Wheeled Walker for Persons with Gait Disorder," *2019 Open Innovations Conference, OI 2019*, pp. 159–164, 2019. DOI: 10.1109/OI.2019.8908169.

[11] C. Werner, G. P. Moustris, C. S. Tzafestas, and K. Hauer, "User-Oriented Evaluation of a Robotic Rollator That Provides Navigation Assistance in Frail Older Adults with and without Cognitive Impairment," *Gerontology*, vol. 64, no. 3, pp. 278–290, 2018, ISSN: 14230003. DOI: 10.1159/000484663.

[12] X. Zhao, Z. Zhu, M. Liu, C. Zhao, Y. Zhao, J. Pan, Z. Wang, and C. Wu, "A Smart Robotic Walker With Intelligent Close-Proximity Interaction Capabilities for Elderly Mobility Safety," *Frontiers in Neurorobotics*, vol. 14, no. October, pp. 1–17, 2020, ISSN: 16625218. DOI: 10.3389/fnbot.2020.575889.

[13] T. Kurai, Y. Shioi, Y. Makino, and H. Shinoda, "Temporal conditions suitable for predicting human motion in walking," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2019-Octob, pp. 2986–2991, 2019, ISSN: 1062922X. DOI: 10.1109/SMC.2019.8913941.

[14] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging LSTMs to Anticipate Actions Very Early," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 280–289, 2017, ISSN: 15505499. DOI: 10.1109/ICCV.2017.39. arXiv: 1703.07023.

[15] J. Paulo, P. Peixoto, and U. J. Nunes, "ISR-AIWALKER: Robotic Walker for Intuitive and Safe Mobility Assistance and Gait Analysis," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 1110–1122, 2017, ISSN: 21682291. DOI: 10.1109/THMS.2017.2759807.

[16] I. S. Weon and S. G. Lee, "Intelligent robotic walker with actively controlled human interaction," *ETRI Journal*, vol. 40, no. 4, pp. 522–530, 2018, ISSN: 22337326. DOI: 10.4218/etrij.2017-0329.

[17] W. C. Cheng and Y. Z. Wu, "A user's intention detection method for smart walker," *Proceedings - 2017 IEEE 8th International Conference on Awareness Science and Technology, iCAST 2017*, vol. 2018-Janua, no. iCAST, pp. 35–39, 2017. DOI: 10.1109/ICAwST.2017.8256477.

[18] S. D. Sierra, J. F. Molina, D. A. Gomez, M. C. Munera, and C. A. Cifuentes, "Development of an Interface for Human-Robot Interaction on a Robotic Platform for Gait Assistance: AGoRA Smart Walker," *2018 IEEE ANDESCON, ANDESCON 2018 - Conference Proceedings*, 2018. DOI: 10.1109/ANDESCON.2018.8564594.

[19] J. H. Park, B. O. Park, and W. G. Lee, "Parametric Design and Analysis of the Arc Motion of a User-Interactive Rollator Handlebar with Hall Sensors," *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 11, pp. 1979–1988, 2019, ISSN: 20054602. DOI: 10.1007/s12541-019-00192-z. [Online]. Available: https://doi.org/10.1007/s12541-019-00192-z.

[20] J. Paulo, P. Peixoto, and U. Nunes, "A novel vision-based human-machine interface for a robotic walker framework," *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, vol. 2015-Novem, pp. 134–139, 2015. DOI: 10.1109/ROMAN.2015.7333590.

[21] C. A. Cifuentes, C. Bayon, S. Lerma, A. Frizera, L. Rodriguez, and E. Rocon, "Converging Clinical and Engineering Research on Neurorehabilitation II," vol. 15, pp. 1451–1455, 2017. DOI: 10.1007/978-3-319-46669-9. [Online]. Available: https://www.springer.com/us/book/9783319466682%7B%5C%%7D0Ahttp://link.springer.com/10.1007/978-3-319-46669-9.

[22] C. Bayón, O. Ramírez, J. I. Serrano, M. D. Castillo, A. Pérez-Somarriba, J. M. Belda-Lois, I. Martínez-Caballero, S. Lerma-Lara, C. Cifuentes, A. Frizera, and E. Rocon, "Development and evaluation of a novel robotic platform for gait rehabilitation in patients with Cerebral Palsy: CPWalker," *Robotics and Autonomous Systems*, vol. 91, pp. 101–114, 2017, ISSN: 09218890. DOI: 10.1016/j.robot.2016.12.015. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2016.12.015.

[23] T. Shen, M. R. Afsar, H. Zhang, C. Ye, and X. Shen, "A 3D Computer Vision-Guided Robotic Companion for Non-Contact Human Assistance and Rehabilitation," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 100, no. 3-4, pp. 911–923, 2020, ISSN: 15730409. DOI: 10.1007/s10846-020-01258-1.

[24] P. Nikdel, R. Shrestha, and R. Vaughan, "The hands-free push-cart: Autonomous following in front by predicting user trajectory around obstacles," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4548–4554, 2018, ISSN: 10504729. DOI: 10.1109/ICRA.2018.8461181.

[25] G. Chalvatzaki, X. S. Papageorgiou, P. Maragos, and C. S. Tzafestas, "Learn to Adapt to Human Walking: A Model-Based Reinforcement Learning Approach for a Robotic Assistant Rollator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3774–3781, 2019, ISSN: 2377-3766. DOI: 10.1109/lra.2019.2929996.

[26] G. Chalvatzaki, P. Koutras, A. Tsiami, C. S. Tzafestas, and P. Maragos, *i-Walk Intelligent Assessment System: Activity, Mobility, Intention, Communication*. Springer International Publishing, 2020, vol. 12538 LNCS, pp. 500–517, ISBN: 9783030668228. DOI: 10.1007/978-3-030-66823-5_30. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-66823-5%7B%5C_%7D30.

[27] J. M. Lopes, J. André, A. Pereira, M. Palermo, N. Ribeiro, J. Cerqueira, and C. P. Santos, "AS-BGo: A Smart Walker for Ataxic Gait and Posture Assessment, Monitoring, and Rehabilitation," *Robotic Technologies in Biomedical and Healthcare Engineering*, pp. 51–86, 2021. DOI: 10.1201/9781003112273-4.

[28] B. Müller, W. Ilg, M. A. Giese, and N. Ludolph, "Validation of enhanced kinect sensor based motion capturing for gait assessment," *PLoS ONE*, vol. 12, no. 4, pp. 14–16, 2017, ISSN: 19326203. DOI: 10.1371/journal.pone.0175813.

[29] D. Berardini, S. Moccia, L. Migliorelli, I. Pacifici, P. di Massimo, M. Paolanti, and E. Frontoni, "Fall detection for elderly-people monitoring using learned features and recurrent neural networks," *Experimental Results*, vol. 1, pp. 1–9, 2020. DOI: 10.1017/exp.2020.3.

[30] Y. Shi, B. Fernando, and R. Hartley, *Action anticipation with RBF kernelized feature mapping RNN*. Springer International Publishing, 2018, vol. 11214 LNCS, pp. 305–322, ISBN: 9783030012489. DOI: 10.1007/978-3-030-01249-6_19. arXiv: 1911.07806. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-01249-6%7B%5C_%7D19.

[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2019, ISSN: 15731405. DOI: 10.1007/s11263-019-01228-7. arXiv: 1610.02391.

[32] M. Baptista-Rios, R. J. Lopez-Sastre, F. Caba Heilbron, J. C. Van Gemert, F. J. Acevedo-Rodriguez, and S. Maldonado-Bascon, "Rethinking Online Action Detection in Untrimmed Videos: A Novel Online Evaluation Protocol," *IEEE Access*, vol. 8, pp. 5139–5146, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2961789. arXiv: 2003.12041.

[33] P. Beckerle, G. Salvietti, R. Unal, D. Prattichizzo, S. Rossi, C. Castellini, S. Hirche, S. Endo, H. B. Amor, M. Ciocarlie, F. Mastrogiovanni, B. D. Argall, and M. Bianchi, "A human-robot interaction perspective on assistive and rehabilitation robotics," *Frontiers in Neurorobotics*, vol. 11, no. MAY, pp. 0–10, 2017, ISSN: 16625218. DOI: 10.3389/fnbot.2017.00024.

[34] L. Lv, J. Yang, D. Zhao, and S. Wang, "A Novel Non-contact Recognition Approach of Walking Intention Based on Long Short-Term Memory Network," *Qian J., Liu H., Cao J., Zhou D. (eds) Robotics and Rehabilitation Intelligence. ICRRI 2020. Communications in Computer and Information Science*, vol. 1335, 2020. DOI: https://doi.org/10.1007/978-981-33-4929-2_2.

[35] S. Page, M. M. Martins, L. Saint-Bauzel, C. P. Santos, and V. Pasqui, "Fast embedded feet pose estimation based on a depth camera for smart walker," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 4224–4229, 2015, ISSN: 10504729. DOI: 10.1109/ICRA.2015.7139781.

[36] S. D. Sierra M., M. Garzón, M. Múnera, and C. A. Cifuentes, "Human–Robot–environment interaction interface for smart walker assisted gait: AGoRA walker," *Sensors (Switzerland)*, vol. 19, no. 13, pp. 1–29, 2019, ISSN: 14248220. DOI: 10.3390/s19132897.

[37] C. Huang, G. Wasson, M. Alwan, P. Sheth, and A. Ledoux, "Shared navigational control and user intent detection in an intelligent walker," *AAAI Fall Symposium - Technical Report*, vol. FS-05-02, pp. 59–66, 2005.

[38] D. Rodriguez-losada, "A Smart Walker for the Blind," *Robotics & automation Magazine*, no. December, pp. 75–83, 2008.

[39] M. Spenko, H. Yu, and S. Dubowsky, "Robotic personal aids for mobility and monitoring for the elderly," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 3, pp. 344–351, 2006, ISSN: 15344320. DOI: 10.1109/TNSRE.2006.881534.

[40] M. F. Jiménez, M. Monllor, A. Frizera, T. Bastos, F. Roberti, and R. Carelli, "Admittance Controller with Spatial Modulation for Assisted Locomotion using a Smart Walker," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 94, no. 3-4, pp. 621–637, 2019, ISSN: 15730409. DOI: 10.1007/s10846-018-0854-0.

[41] C. Canuto, P. Moreno, J. Samatelo, R. Vassallo, and J. Santos-Victor, "Action anticipation for collaborative environments: The impact of contextual information and uncertainty-based prediction," *Neurocomputing*, vol. 444, no. xxxx, pp. 301–318, 2021, ISSN: 18728286. DOI: 10.1016/j.neucom.2020.07.135. arXiv: 1910.00714. [Online]. Available: https://doi.org/10.1016/j.neucom.2020.07.135.

[42] A. Yeaser, J. Tung, J. Huissoon, and E. Hashemi, "Learning-Aided User Intent Estimation for Smart Rollators," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020-July, pp. 3178–3183, 2020, ISSN: 1557170X. DOI: 10.1109/EMBC44109.2020.9175610.

[43] T. M. Chalen and B. Vintimilla, "Towards Action Prediction Applying Deep Learning," *2019 IEEE Latin American Conference on Computational Intelligence, LA-CCI 2019*, pp. 1–3, 2019. DOI: 10.1109/LA-CCI47412.2019.9037051.

[44] H. B. Zhang, Y. X. Zhang, B. Zhong, Q. Lei, L. Yang, J. X. Du, and D. S. Chen, "A comprehensive survey of vision-based human action recognition methods," *Sensors (Switzerland)*, vol. 19, no. 5, pp. 1–20, 2019, ISSN: 14248220. DOI: 10.3390/s19051005.

[45] A. Jalal, Y. H. Kim, Y. J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recognition*, vol. 61, pp. 295–308, 2017, ISSN: 00313203. DOI: 10.1016/j.patcog.2016.08.003. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2016.08.003.

[46] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9911 LNCS, pp. 203–220, 2016, ISSN: 16113349. DOI: 10.1007/978-3-319-46478-7_13. arXiv: 1604.05633.

[47] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9917–9926, 2019, ISSN: 10636919. DOI: 10.1109/CVPR.2019.01016.

[48] C. Canuto, P. Moreno, J. Samatelo, R. Vassallo, and J. Santos-Victor, "Action anticipation for collaborative environments: The impact of contextual information and uncertainty-based prediction," *Neurocomputing*, vol. 444, no. xxxx, pp. 301–318, 2021, ISSN: 18728286. DOI: 10.1016/j.

neucom.2020.07.135. arXiv: 1910.00714. [Online]. Available: https://doi.org/10.1016/j.neucom.2020.07.135.

[49] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973, ISSN: 00315117. DOI: 10.3758/BF03212378.

[50] F. Han, B. Reily, W. Hoff, and H. Zhang, "Space-time representation of people based on 3D skeletal data: A review," *Computer Vision and Image Understanding*, vol. 158, pp. 85–105, 2017, ISSN: 1090235X. DOI: 10.1016/j.cviu.2017.01.011. arXiv: 1601.01006.

[51] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 7745–7754, 2019, ISSN: 10636919. DOI: 10.1109/CVPR.2019.00794. arXiv: 1811.11742.

[52] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H. P. Seidel, W. Xu, D. Casas, and C. Theobalt, "VNect: Real-time 3D human pose estimation with a single RGB camera," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–13, 2017, ISSN: 15577368. DOI: 10.1145/3072959.3073596. arXiv: 1705.01583.

[53] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. van Gool, "Temporal segment networks: Towards good practices for deep action recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9912 LNCS, pp. 20–36, 2016, ISSN: 16113349. DOI: 10.1007/978-3-319-46484-8_2. arXiv: 1608.00859.

[54] C. Rodriguez, B. Fernando, and H. Li, *Action anticipation by predicting future dynamic images*. Springer International Publishing, 2019, vol. 11131 LNCS, pp. 89–105, ISBN: 9783030110147. DOI: 10.1007/978-3-030-11015-4_10. arXiv: 1808.00141. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-11015-4%7B%5C_%7D10.

[55] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 1, no. January, pp. 568–576, 2014, ISSN: 10495258. arXiv: 1406.2199.

[56] S. T. H. Shah and X. Xuezhi, "Traditional and modern strategies for optical flow: an investigation," *SN Applied Sciences*, vol. 3, no. 3, pp. 1–14, 2021, ISSN: 25233971. DOI: 10.1007/s42452-021-04227-x. [Online]. Available: https://doi.org/10.1007/s42452-021-04227-x.

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015. arXiv: 1409.1556.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, ISSN: 10636919. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

[59] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17, San Francisco, California, USA: AAAI Press, 2017, pp. 4278–4284.

[60] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[61] R. De Geest and T. Tuytelaars, "Modeling temporal structure with LSTM for online action detection," *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol. 2018-Janua, pp. 1549–1557, 2018. DOI: 10.1109/WACV.2018.00173.

[62] S. Guo, L. Qing, J. Miao, and L. Duan, "Action prediction via deep residual feature learning and weighted loss," *Multimedia Tools and Applications*, vol. 79, no. 7-8, pp. 4713–4727, 2020, ISSN: 15737721. DOI: 10.1007/s11042-019-7675-4.

[63] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 98–106, 2016, ISSN: 10636919. DOI: 10.1109/CVPR.2016.18. arXiv: 1504.08023.

[64] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," *British Machine Vision Conference 2017, BMVC 2017*, 2017. DOI: 10.5244/c.31.92. arXiv: 1707.04818.

[65] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 2921–2929, 2016, ISSN: 10636919. DOI: 10.1109/CVPR.2016.319. arXiv: 1512.04150.

[66] I. Farkhatdinov, N. Roehri, and E. Burdet, "Anticipatory detection of turning in humans for intuitive control of robotic mobility assistance," *Bioinspiration and Biomimetics*, vol. 12, no. 5, 2017, ISSN: 17483190. DOI: 10.1088/1748-3190/aa80ad.

[67] R. Girdhar, J. Joao Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 244–253, 2019, ISSN: 10636919. DOI: 10.1109/CVPR.2019.00033. arXiv: 1812.02707.

[68] D. Liu, Y. Wang, and J. Kato, "Supervised spatial transformer networks for attention learning in fine-grained action recognition," *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 4, no. Visigrapp, pp. 311–318, 2019. DOI: 10.5220/0007257803110318.

[69] Y. Qiao, W. Cui, and T. Shi, "LaM-2SRN: A method which can enhance local features and detect moving objects for action recognition," *IEEE Access*, vol. 8, pp. 192 703–192 712, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3032533.

[70] H. Wu, X. Ma, and Y. Li, "Convolutional Networks with Channel and STIPs Attention Model for Action Recognition in Videos," *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2293–2306, 2020, ISSN: 19410077. DOI: 10.1109/TMM.2019.2953814.

[71] A. Kozlov, V. Andronov, and Y. Gritsenko, "Lightweight network architecture for real-time action recognition," *Proceedings of the ACM Symposium on Applied Computing*, pp. 2074–2080, 2020. DOI: 10.1145/3341105.3373906. arXiv: 1905.08711.

[72] R. Moreira, J. Alves, A. Matias, and C. P. Santos, "Smart and Assistive Walker – ASBGo: Rehabilitation Robotics: A Smart– Walker to Assist Ataxic Patients," in *Robotics in Healthcare. Advances in Experimental Medicine and Biology.* Springer Nature Switzerland AG, 2019, pp. 37–68, ISBN: 9783319326696. DOI: 10.1007/978-3-319-32669-6.

[73] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," no. November, 2012. arXiv: 1212.0402. [Online]. Available: http://arxiv.org/abs/1212.0402.

[74] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, no. PART 3, pp. 410–424, 2014, ISSN: 16113349. DOI: 10.1007/978-3-319-10578-9_27.

[75] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Histogram of Oriented Principal Components for Cross-View Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2430–2443, 2016, ISSN: 01628828. DOI: 10.1109/TPAMI.2016.2533389. arXiv: 1409.6813.

[76] M. Palermo, S. Moccia, L. Migliorelli, E. Frontoni, and C. P. Santos, "Real-time human pose estimation on a smart walker using convolutional neural networks," *Expert Systems with Applications*, vol. 184, pp. 1–15, 2021, ISSN: 09574174. DOI: 10.1016/j.eswa.2021.115498. arXiv: 2106.14739.

[77] B. P. O'Callaghan, E. P. Doheny, C. Goulding, E. Fortune, and M. M. Lowery, "Adaptive gait segmentation algorithm for walking bout detection using tri-axial accelerometers," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020-July, pp. 4592–4595, 2020, ISSN: 1557170X. DOI: 10.1109/EMBC44109.2020.9176460.

[78] G. D. M. Jenkin, *Computational principles of mobile robotics*, 2ed. Cambridge University Press, 2010, ISBN: 9780521871570; 0521871573; 9780521692120; 0521692121. [Online]. Available: libgen.li/file.php?md5=fcf0f4e452ec55632b18c18863bcb6fa.

[79] J. Figueiredo, S. P. Carvalho, D. Goncalve, J. C. Moreno, and C. P. Santos, "Daily Locomotion Recognition and Prediction: A Kinematic Data-Based Machine Learning Approach," *IEEE Access*, vol. 8, pp. 33 250–33 262, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2971552.

[80] A. Pachi and T. Ji, "Frequency and velocity of people walking," *The Structural engineer*, vol. 83, 2005.

[81] D. Sun, X. Yang, M. Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. D, pp. 8934–8943, 2018, ISSN: 10636919. DOI: 10.1109/CVPR.2018.00931. arXiv: 1709.02371.

[82] G. Chen, P. Chen, Y. Shi, K. Hsieh, B. Liao, and S. Zhang, "Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks," 2019. arXiv: 1905.05928. [Online]. Available: https://arxiv.org/pdf/1905.05928.pdf.

[83] Q. Ji, J. Huang, W. He, and Y. Sun, "Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images," *Algorithms*, vol. 12, no. 3, pp. 1–12, 2019, ISSN: 19994893. DOI: 10.3390/a12030051.

[84] S. Mehta, E. Mercan, J. Bartlett, D. Weaver, J. G. Elmore, and L. Shapiro, "Y-Net: Joint segmentation and classification for diagnosis of breast biopsy images," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11071 LNCS, pp. 893–901, 2018, ISSN: 16113349. DOI: 10.1007/978-3-030-00934-2_99. arXiv: 1806.01313.

[85] O. Ronneberger, P. Fischer, and T. Brox, "U-Net : Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 9351 LNCS, Springer, 2015, pp. 234–241, ISBN: 9783319245744. DOI: 10.1007/978-3-319-24574-4. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.

[86] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," no. May 2014, pp. 248–255, 2010. DOI: 10.1109/cvpr.2009.5206848.

[87] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814, no. Figure 1, pp. 1575–1585, 2020. arXiv: 2002.05709.

[88] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, no. Figure 2, pp. 6706–6716, 2020, ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00674. arXiv: 1912.01991.