Deep Leg Tracking by Detection and Gait Analysis in 2D Range Data for Intelligent Robotic Assistants

Danai Efstathiou[1], Georgia Chalvatzaki[2], Athanasios Dometios[1],
Dionisios Spiliopoulos[1] and Costas S. Tzafestas[1]

***Abstract*— Online human leg tracking and gait analysis are crucial functionalities for mobility assistant robots, like intelligent walkers. Usually, such walkers are equipped with various sensors for the extraction of human-related features for adaptive human-robot interaction and assistance. We treat the gait detection problem jointly, presenting a novel method for detecting and recognizing gait features from 2D range data produced by a laser sensor mounted on a robotic walker. We propose an effective Convolutional Neural Network (CNN) as a powerful feature extractor for detecting the user's leg centers in range data represented as occupancy grid maps. We couple the CNN with a Long Short Term Memory (LSTM) network for learning the legs' motion temporal dynamics while walking, improving the prior detection, and providing better leg occlusion handling. Moreover, we perform gait analysis by recognizing gait phases over both legs by feeding the leg tracking output to a subsequent LSTM. Our proposed lightweight framework has been trained and tested on real patients-data. The presented experimental results show our method's efficiency in providing accurate detections compared to state-of-the-art and application to an online system due to its high frequency, making it a competitive method for gait detection on robotic mobility assistants.**

I. INTRODUCTION

Functional walking is an integral part of every person's daily life. When mobility disabilities emerge, usually with age, depriving a human of a vital ability taken for granted, they affect the individual physically and emotionally [2]. A mobility-impaired person can also be prone to fall incidents, which can easily cause injuries that could provoke further, often more severe, problems. Conventional walkers and canes have played a significant role in assisting mobility-impaired humans, without unfortunately eliminating falling incidents, nor being easily adaptable to every patient's specific needs [1], [6]. The current technological advances, especially in robotics, can help every individual retain normalcy in their everyday life, and their independence and self-esteem [3]–[5]. This work's motivation is to equip such a walker with technologies to assess the individual's needs and adapt to assist as optimally as possible. A context-aware robotic walker could improve on patient-supporting, guiding, fall prevention, or even rehabilitation [8].

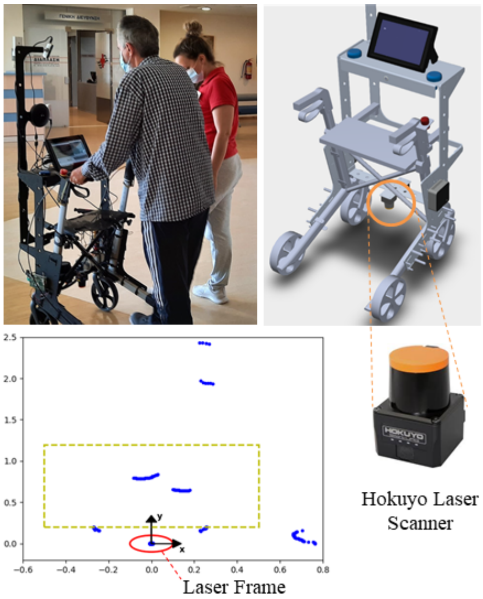

Fig. 1: **Up-Left:** A patient using an intelligent robotic walker. **Up-Right:** Sketch of the walker design. A Hokuyo laser scans the walking area at a height ∼ 35cm above the ground. **Down:** 2D range data (blue) from the laser scanner, whose origin is marked by a red circle. Due to the existence of obstacles other than the patient's legs, only the laser points lying in the bounding box (yellow) are considered for leg detection and gait analysis.

Among all different functionalities, there is one necessary feature that a robotic walker has to be equipped with: a gait tracking and analysis mechanism. Leg tracking refers to the accurate estimation of human legs' position throughout time. The way a human walks can suggest much about them, such as disabilities [9], [10], an inclination to fall [27], etc. Therefore, an intelligent walker with efficient tracking and gait analysis system can also be employed for more sophisticated functionalities that correspond to the user's needs. However, the proximity of the user to the robot during the supportive actions of walking (Fig. 1), along with the pathological aspects that alter the gait patterns and walking frequencies, make the leg tracking very challenging, while the gait analysis task from partial 2D observations is increasingly demanding – in close proximity depth cameras are performing poorly, fast 3d range-scanners are prohibitively expensive and elderly patients are skeptical to wearable sensors. Notably, only a few works in the literature consider gait tracking using range data for robotic walkers [22].

*This research has been partially funded by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (i-Walk project code:T1EDK-01248/ MIS: 5030856). G. Chalvatzaki's work is funded by the EN DFG Programme iROSA (CH 2676/1-1).

[1] Electrical and Computer Engineering, National Technical University of Athens, Greece. danaiefst@gmail.com, dennisspiliopoylos@gmail.com, athdom@mail.ntua.gr, ktzaf@cs.ntua.gr

[2] Intelligent Robotics for Assistance group, Technische Universitat Darmstadt, Germany georgia@robot-learning.de

Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021

This paper presents a novel deep learning framework for Leg Tracking by detection and Gait Analysis from 2D range data (LTGADnet). The proposed approach uses a CNN for leg detection, followed by an LSTM network for exploiting temporal information in walking and revising CNN's detections in challenging situations like leg occlusions. Finally, a second LSTM is used to extract the high-level temporal interaction of the legs, which can provide evidence about the occurring gait phase, resulting in a real-time gait analysis system. Our key contribution is a highly accurate leg tracking by detection method, thanks to the deep feature extraction, and an implementation that can be deployed as an off-the-shelf leg tracking method for any robotic mobility assistant, due to its effectiveness and high-frequency, without the need for extra calibrations or thresholds. We provide experimental evidence about our algorithmic solution using our novel Leg Tracking and Gait Analysis from 2D range data (LTGAD) database, which comprises data and annotations from real patients using mobility devices, and we compare with the most recent algorithm that was developed for such applications [22], showcasing the performance increase of our method.

II. Related Work

A. Leg Tracking Algorithms

Human leg tracking has been a popular subject on robotic applications, mostly for human detection, tracking, and following. Data derived from various sensors, including lasers, cameras, markers, etc., are used and often fused to estimate the position of human legs in sequential time frames. A fusion of RGB-marker and IMU data has been proposed for leg detection, combined with an extended Kalman filter for leg tracking [14]. Biometric data have also been used for human detection [15].

As most mobile robots are equipped with a 2D Laser Imaging Detection and Ranging (LIDAR) sensor, due to their reliability and affordability, there has been plenty of proposed work on learning algorithms for processing the laser sensor data for human detection and tracking. Many of those methods, either used a leg pattern recognition scheme [11] [12], boosted classifiers [16] or clustering methods [17] [13] and Kalman filters for the tracking part. Leg tracking and gait analysis using two particle filters, one per leg, and probabilistic data association with an interactive multiple model scheme have also been proposed by [22]. Such implementations, however, have a high computational cost, and thus the high frequency of modern laser scanners cannot be fully exploited.

On the other hand, deep learning methods have also been considered in human tracking due to their scalability and fast inference. The use of CNN is presented in [18] for detecting people in crowds from range data. A U-Net architecture [21], commonly used for biomedical image segmentation, has also been proposed in [20]. These methods, though, perform person detection and do not consider learning the human legs' dynamic motion that can be exploited for tracking, which is crucial for further gait analysis.

B. Deep Learning Detection

Deep neural networks have achieved exceptional results in object detection in general. The widely popular YOLO architecture [23] can detect multiple objects and suggest a bounding box for every one of them.

For this architecture to be used in our work, a specific and extensive dataset including bounding boxes would have to be created, which should not be mandatory for our task. The leg tracking problem can not be solved by a single frame object detection neural network, as occlusions can temporarily make a leg invisible. Due to the necessity for an assistive walker to have a consistent awareness of the patient's leg position, a neural network architecture with temporal awareness is called for. For this purpose, two ways of extending a CNN were considered.

The first one was the use of a Long Short Term Memory (LSTM). In [24], for the exploitation of the temporal information that a video provides, a neural network with a Single Shot Multibox Detector (SSD), followed by two LSTM layers for object detection on video streams, was presented. Similarly, in [25], an LSTM layer was used after a YOLO for improving detection performance.

The second one was the use of a Temporal Convolutional Network (TCN). As TCNs have shown promising results and often a better performance than LSTMs, a Dilated TCN layer, similar to the one proposed in [26] for motion capture, was added after the CNN. This TCN uses exponentially increasing dilation between consecutive 1D convolutional layers for associating the features between successive data.

Our proposed architecture includes two neural networks, one for leg detection and tracking and one for gait analysis. It uses a CNN for leg detection and an LSTM for handling occlusions. We also propose the use of an LSTM for gait analysis. Furthermore, computer-generated (CG) data have been developed to train both networks, proving to improve performance significantly. The neural networks developed to show high accuracy on both the leg detection and the gait phase extraction. Moreover, due to the use of a laser scan and neural networks, our framework has a low computational cost and can be easily applied to any walker.

III. Problem Statement

The problem we aim to solve is to perform accurate, efficient, and robust leg tracking and gait analysis on a patient using a smart walker equipped with a 2D LIDAR. Given readings of the laser sensor at each time frame t, i.e., distances from the laser to detected objects, our goal is twofold: (1) to find the relative coordinates of the centers $(x_r[t], y_r[t])$ and $(x_l[t], y_l[t])$ of the patient's right and left leg w.r.t. the laser's position and (2) given these centers, to estimate the patient's gate state. These states represent specific phases of walking, as defined in gait analysis literature. The gait states we are considering in this work are shown in Table I. These gait states are a part of a Markov chain, where every state at every time frame can either stay the same or transition to the next one in the chain.

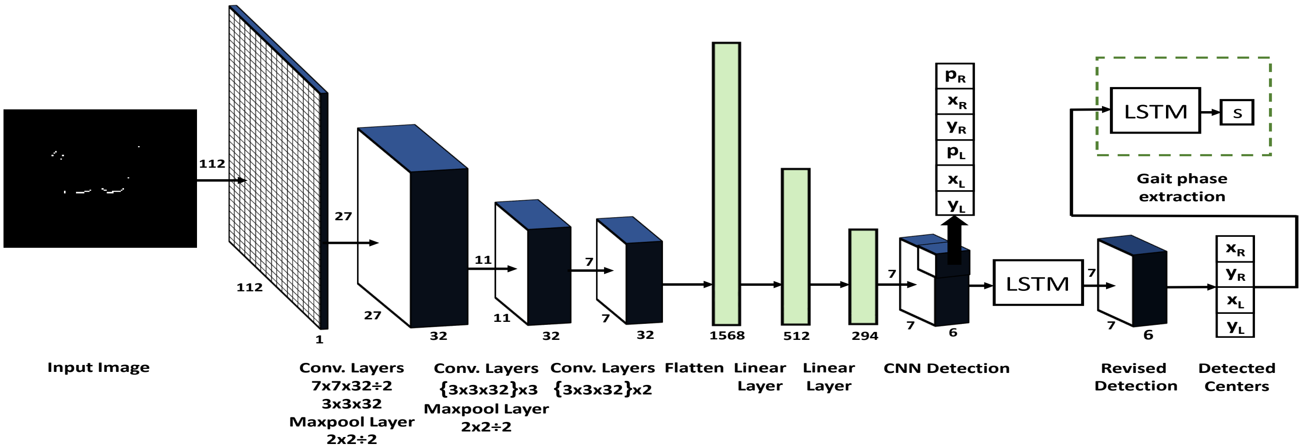

Fig. 2: The architecture of our proposed framework for leg tracking by detection composed of Convolutional and Linear Layers followed by an LSTM. The output of the leg detection is fed into a single LSTM that recognizes the gait phase according to the one-hot key encoding of Table I.

TABLE I: Gait State representation as in [22] and their one-hot key encoding.

State Code	One-Hot Key	Code Name	Definition
s_1	[1, 0, 0, 0]	LDS	Left Double Support
s_2	[0, 1, 0, 0]	LS/RW	Left Swing/Right Stance
s_3	[0, 0, 1, 0]	RDS	Right Double Support
s_4	[0, 0, 0, 1]	RS/LW	Right Swing/Left Stance

IV. Proposed Method

A. Network Input

The network's input is an occupancy grid derived from the 2D laser data. The distances and angles of the objects provided by the laser are converted to (x, y) coordinates in the Cartesian system relative to the position of the laser. Only the objects lying in a prespecified bounding box, as explained in section IV, are considered valid (the human interacting with the robotic assistant should be in specific proximity to the robot to start the interaction). We specify an occupancy grid of size 112×112, in which 1 indicates that an object was detected in this grid cell, while 0 refers to no object detection. This occupancy map is fed into our neural network, which we describe in the following paragraph.

B. LTGADnet Architecture

Our proposed LTGADnet framework is depicted in Fig. 2. The tracking by detection architecture comprises CNNs, linear layers, and finally, an LSTM layer. We took inspiration from the YOLO framework to design our architecture, as it is a successful neural network for real-time object detection. However, our problem lacks dimensionality compared to the object detection problems that YOLO architectures tackle. Hence, we have redesigned our architecture to propose a novel tracking by detection framework for the dynamic problem of leg tracking. Our architecture takes as input an image of size 112×112, the occupancy grid, and it segments the image in a 7×7 grid, named the detection grid. For each grid cell, it outputs a vector containing for every leg (1) the probability that the leg was found, (2) the x of the center of the leg, and (3) the y of the center of the leg. From the 6 outputs, the first three always correspond to the right leg and the next three to the left.

The CNN layers are the following: $7 \times 7 \times 32 \div 2$ Convolution (e.g. kernel size 7, number of channels 32, stride 2), $3 \times 3 \times 32$ Convolution, $2 \times 2 \div 2$ Max Pooling, $3 \times 3 \times 32$ Convolution, $3 \times 3 \times 32$ Convolution, $3 \times 3 \times 32$ Convolution, $2 \times 2 \div 2$ Max Pooling, $3 \times 3 \times 32$ Convolution, $3 \times 3 \times 32$ Convolution. The output of these layers is of size $7 \times 7 \times 32$ and is flattened, in order to be fed to the linear layers.

Then, two linear layers follow, of size 1568×512 and 512×294, where $294 = 7 \times 7 \times 6$. This is the detection output of the CNN. Also, every single layer is followed by batch normalization and ReLU activation function. For training, we applied dropout 0.5 after the first linear layer.

Lastly, a one-layer LSTM receives as input a sequence of detection frames, namely the output of the last linear layer for consecutive frames, and produces the corrected detections for these frames, which have the same size as the LSTM's input. The LSTM's output is computed as described in [29]. From the output of the LSTM, we extract the legs' centers by picking the coordinates of the grid cell with the highest confidence probability and summing them with the relative coordinates produced by that cell. Finally, these 4 coordinates of the leg centers are given as input to a five-layer LSTM. The LSTM gives an output of size 4, which represents the one-hot encoding of the estimated gait state. For the training of this LSTM, dropout 0.3 was applied between the layers.

C. Training

For training the leg tracking and the gait analysis networks, we used the following loss functions:

$$\mathit{Confidence\ Loss} = \sum_{i=1}^{7}\sum_{j=1}^{7}\sum_{k=1}^{2}(C_{ijk} - \hat{C}_{ijk})^2$$

$$\mathit{Detection\ Loss} = \sum_{i=1}^{2}(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

$$\mathit{Loss} = \mathit{Confidence\ Loss} + \alpha \cdot \mathit{Detection\ Loss} \tag{1}$$

where C_{ijk} is the expected confidence of the grid cell ij (1 if the cell contains leg k and 0 for the rest), $\hat{C}_{ijk}$ the output confidence of the grid cell ij for leg k, x_i, y_i the position of leg i in the grid and $\hat{x}_i, \hat{y}_i$ the position of leg i in the grid as a result of the output.

In training, $\alpha = 5$ proved to achieve the best results. After some experimentation on the batch size with no obvious impact, batch size 32 was used. Adam was used as an optimizer, with a learning rate starting at 10^{-4} and decaying at 10^{-5} and 10^{-6} at epochs 25 and 50 respectively. The leg tracking network was trained for 100 epochs. The gait analysis network was trained with Cross-Entropy loss and batch size also 32. Adam was used as the optimizer, while the learning rate began at 10^{-3} and decayed at 10^{-4}, 10^{-5} and 10^{-6} at epochs 10, 20 and 40 respectively.

For both the training of the leg tracking and the gait analysis networks, a technique similar to early stopping was applied. Every time the loss at the validation set decreased, we saved the current model. The last model to be saved was the one to be finally used.

D. Inference

LTGADnet comprises a total of 1704694 parameters (1703958 for tracking and only 736 for gait analysis), making it a very lightweight solution, allowing it, e.g., when running on a GeForce GTX 1060 6GB GPU, to provide real-time leg detection and gait states predictions, easily following the 40Hz frequency of our laser sensor, while it is able to perform at a frequency of one order of magnitude higher than the sensor.

V. Experimental Results

A. Dataset

The dataset[1] used for training, validating, and testing the LTGADnet was created by the data produced by a smart walker maneuvered by real patients and is the same dataset used for the IMM-PF in [22]. All patients were over 65 years old. Our dataset comprises annotated data from 8 experiments realized by different patients. Crucially, the ground truth (GT) labels for training the network were extracted by a VICON motion capture system that was used during the experiments, with markers placed on specific areas of the patient's body. The process of extracting GT leg trajectories and identifying the GT gait states has been thoroughly described in [22].

The collection of real experimental data from patients is very challenging. Our dataset consists of approx. 33000 frames. Therefore, for improving the performance of LTGADnet, we applied data augmentation. We have used mirroring and extensive shifting of the occupancy grids and the leg centers. Moreover, we have included CG data for training the network. For the fabrication of these data, we "imitated" the walker set up, assuming that the laser sensor lies at the origin $(0, 0)$, as shown in Fig. 1 while simulating hypothetical patient's legs. First, we designed two circles representing the legs. We assumed a sinusoidal motion on the gait direction and another on its perpendicular direction, but with a much smaller amplitude, frequency, and a slightly irregular phase increase at each time step. The gait direction also changes through time, creating virtual turning moves and thus occlusions. The frequencies follow the analysis of normal human gait, as found in [28]. The points in which the laser beams and the leg circles intersect, if they exist, are calculated and are shifted on the laser beam direction using a Gaussian noise with zero mean and standard deviation one, suitable to simulate the nominal error of the real sensor. We have also simulated the faulty trailing patterns, which are often present in 2D laser data. After a selective choice of the annotated data and our extensive data augmentation, we end up with approx. 210000 frames for training, validation, and testing, which are also included in our public LTGAD database. We used 6 real experiments for training, 1 for validation and 1 for testing, as well as the CG data for training, which are 58% of the augmented real data.

B. Evaluation metrics and analysis

For the leg tracking network, we use as metrics the mean, max, and median euclidean distance between the detected centers and the annotated ones. The max and median distances are provided, as rare occasions of one-frame faulty detections from the CNN can skew the mean distance upwards. For the gait analysis network, we calculated the overall accuracy of the gait state detection and the Recall, Precision, and F1 scores. For the latter three metrics, we use the weighted mean over all gait states, as there exists an inherent class imbalance in the data, with stance states (s_1, s_3) covering only $\sim 38\%$ of the dataset.

We compare our results with the ones in [22], which is the state-of-the-art leg tracking and gait analysis algorithm and the only one that tackles the same problem. We could not compare with other works, such as [18], [20], as they perform person tracking instead of individual leg tracking.

C. Ablation Study

To justify the architectural design choices of LTGADnet of Fig. 2, we present an ablation study in Table III. We specifically focus on ablating the tracking by detection part of the framework, where various design choices had to be made. The different architectures that we tested are compared based on the mean and the max euclidean distances between the GT leg centers and the produced ones in the validation set. The different designs that were tested are (a) the CNN of Fig. 2 noted as `CNN7`, (b) a CNN without the last Convolutional Layer, resulting to an output of size $9 \times 9 \times 32$, which is then fed to a first linear layer of size 2592×512 noted as `CNN9`, (c) a full network (as in Fig. 2) with a one-layer LSTM trained with loss function (1) with $\alpha = 5$ plus an additional leg association loss with a weight $\beta = 0.1$, noted as `LSTM1assoc`, (d) a full network (as in Fig. 2) with a one-layer LSTM trained with loss function (1) and $\alpha = 5$, noted as `LSTM1`, (e) a full network (as in Fig. 2), but with a two-layer LSTM trained with loss function (1) and $\alpha = 5$,

[1] https://robotics.ntua.gr/ltgad/

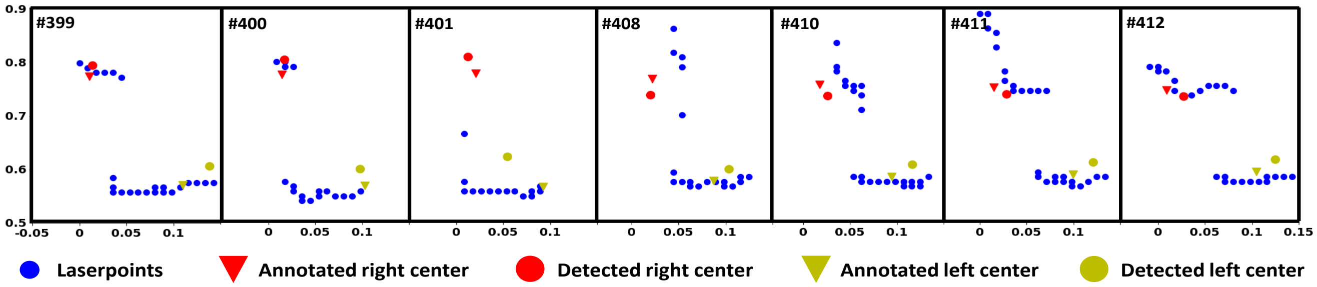

Fig. 3: An example of the results of our architecture, when an occlusion occurs. The plots shown represent the results of the network over consecutive frames. It is displayed that although in frames #400-#410 there is a scarcity of laser points on the right leg, making it invisible, the network is capable of keeping track of it due to the use of the LSTM.

TABLE II: Leave-one-out validation results

Leg Tracking									
Metric \ Experiment	1	2	3	4	5	6	7	8	mean
Mean (cm)	2.42	2.04	3.28	4.11	4.57	2.46	3.26	3.68	3.23
Max (cm)	9.63	9.87	22.16	58.32	14.59	7.79	30.31	42.95	24.45
Median (cm)	2.21	1.81	2.60	2.95	3.74	2.18	2.75	3.31	2.69

Gait Analysis									
Metric \ Experiment	1	2	3	4	5	6	7	8	mean
Accuracy (%)	75.60	71.56	68.85	72.74	69.27	63.17	69.36	75.89	70.805
Precision (%)	78.88	71.36	68.87	74.80	70.76	67.85	69.38	77.50	72.425
Recall (%)	75.60	71.57	68.85	72.74	69.27	63.17	69.37	75.89	70.8075
F1 score (%)	76.57	71.28	67.62	72.10	68.68	61.94	69.21	76.24	70.455

TABLE III: Ablation Study

Architectures	Mean distance (cm)	Max distance (cm)
`CNN7`	3.67	37.05
`CNN9`	4.27	32.03
`LSTM1`	**3.16**	**15.26**
`LSTM2`	3.15	19.21
`TCN`	3.47	40.10
`LSTM1assoc`	3.21	13.51
`TCNassoc`	3.87	48.80

noted as `LSTM2`, (f) a full network (as in Fig. 2) with a Dilated TCN with max dilation 8, trained with loss function (1) and $a = 5$, noted as `TCN`, (g) a full network with TCN, with max dilation 8, trained with loss function (1) plus the leg association loss and parameters $\alpha = 5, \beta = 0.1$, noted as `TCNassoc`.

Inspecting the results in Table III, we notice that overall an architecture based on LSTMs provides more accurate results than TCNs. TCNs' performance depends on the training batch size and has a high memory cost as it needs long sequences of data. Our dataset of real patient walking instances limits us to short batch training, in which the LSTM with its efficient memory handling beats TCN's performance. Moreover, the 2-layer LSTM (`LSTM2`) does not improve the detection performance considerably. Notably, we found that the addition of the association error in `LSTMassoc`, `TCNassoc` does not seem to have a great effect on the performance, but it rather complicates the training objective providing slightly worse results. The best architecture was the one with 1-layer LSTM (`LSTM1`), as the one depicted in Fig. 2, trained with loss function (1). This shows that the combination of the superior feature extraction of CNNs, when combined with the ability of LSTMs to encode temporal dynamics and predict the evolution of the targets over time, is an effective method for challenging dynamic tasks, including the leg tracking by detection from 2D range data. Our lightweight architecture contains 1703958 parameters, making it very efficient for real-time performance on any mobile robotic assistant, like the one in Fig.1.

D. Validation and Testing

We further validated our proposed LTGADnet using a leave-one-out cross-validation strategy. In this case, for every training session, we exclude one experiment and use it for testing, while another one is used for validation. The results of the cross-validation are shown in Table II. We performed tests both for tracking and gait analysis. Regarding the leg tracking, our results show the overall good performance of our method across different combinations of train/tests, indicating the generalization ability of LTGADnet in tracking legs of real patients, who suffer from various pathologies that affect their walking performance (hence, variable dynamics to be learned by our network). Notably, we report much more accurate leg detection than [22], whose probabilistic approach delivered an average mean distance error of 6.69cm while requiring more resources, as the particle filter is computationally more expensive, and therefore challenging to be compatible with the laser scanner's frame rate (40Hz) for online performance. Our lightweight deep method achieves an impressive 52% improvement over the state-of-the-art in tracking accuracy, making it an efficient leg tracking method to be employed in any mobility assistant robot equipped with a 2D range sensor.

Moreover, we report our results on the Gait Analysis problem. Here, our results are found sufficient yet not optimal. Our average accuracy over all gait phases and all tests is $\sim 71\%$ with an F1-score of $\sim 70\%$. This result stems from two major parameters. (i) Pathological walking comprises great variability in the different gait phases [28]. The representation of the gait phases as in Table I, imposes the difficulty of recognizing the joint state of the legs, needing difficult dynamics to be extracted from 2D data. Note that, for example, a swing phase initiates when the toe leaves the ground [28], which is very difficult to be captured by the 2D representation of the leg movement (especially when detecting points on the tibia). Moreover, the DS phases are very short subphases, though crucial for transitioning in walking, but tough to be detected. In our dataset, only 19% of the instances belong to the LDS and 18.6% to the RDS phase, making our dataset highly imbalanced, reflected in our results. (ii) Our dataset seems to be small for such a demanding task. To understand how variable gait dynamics are, we should consider that a normal gait cycle is usually composed of 60% stance and 40% approx. [28], while in our dataset, subject 1 has on average 62.60% stance and 37.40% swing, while subject 6 has 73.14% stance and 26.86% swing per gait cycle (subjects picked randomly). Note that we only trained/tested on instances of walking activity, as such detection is possible in our overall integrated intelligent system, showcased in [1], and thus our results cannot be directly compared with the ones in [22]. The high variability in the duration of the phases, together with the high sensitivity over the potentially different pathological walking dynamics, demands a much broader dataset than the one we are experimenting with.

VI. CONCLUSIONS

We proposed LTGADnet, a novel, lightweight deep learning architecture for efficient human leg tracking and gait analysis from 2D range data. LTGADnet can be employed as an off-the-shelf method to any mobility assistant robot equipped with a laser sensor that scans the user's walking area. Our network architecture comprises a CNN followed by an LSTM to effectively detect the user's legs, using as input an occupancy grid representation of the range data. The superior feature extraction power of convolutions combined with the ability of LSTMs in learning temporal dependencies offers the possibility to learn the motion dynamics of walking, tracking both legs, and even dealing with challenging cases of leg occlusions e.g., during turning. Moreover, we feed the tracking network's output into a simple LSTM layer that learns a high-level classification of the human gait phases. Our experimental results demonstrated the improved performance of LTGADnet in the tracking by detection problem w.r.t. the costly probabilistic state-of-the-art method. While we got sufficient performance for the gait analysis problem, we believe that the lack of a bigger dataset hinders our method's performance. In the future, we will seek to collect more real-world data both from healthy subjects and patients. We will also explore self-supervised learning methods to alleviate the need for annotations that are admittedly difficult to produce.

REFERENCES

[1] Chalvatzaki, G, Koutras, P, Tsiami, A, Tzafestas, C. & Maragos, P. i-Walk Intelligent Assessment System: Activity, Mobility, Intention, Communication, ECCV 2020.

[2] Tough, H, Siegrist, J. & Fekete, C. Social relationships, mental health and wellbeing in physical disability: a systematic review, BMC Public Health 2017.

[3] Martins, M, P. Santos, C, Frizera-Neto, A. & Ceres, R. Assistive mobility devices focusing on Smart Walkers: Classification and review, Robotics and Autonomous Systems 2012 60(4).

[4] Chaparro Cárdenas, S. & Lozano-Guzmán et al. A review in gait rehabilitation devices and applied control techniques, Disability and Rehabilitation: Assistive Technology 2018 13.

[5] Bradley, S, & R. Hernandez, C. Geriatric assistive devices, American Family Physician 2011 84(4).

[6] Riel, K, Hartholt, K, Panneman, M. & Patka et al. Four-wheeled walker related injuries in older adults in the Netherlands, Inj Prev. 2014 20(1).

[7] Schiariti, V. & Pelligra, G. Developmental Milestones of Assistive Technology: From Wood Walking Sticks to Virtual Reality, Paediatrics Child Health 2015 20(5).

[8] Koumpouros, Y, Toulias, T, Tzafestas, C. & Moustris, G. Assessment of an intelligent robotic walker implementing navigation assistance in frail seniors, Technology and Disability 2020 32(3).

[9] Amboni, M, Barone, P. & Hausdorff, J. Cognitive Contributions to Gait and Falls: Evidence and Implications, Mov Disord. 2013 28(11).

[10] Beauchet, O. et al. Gait analysis in demented subjects: Interests and perspectives, Neuropsychiatr Dis Treat. 2008 4(1).

[11] Bellotto, N. & Hu, H. Multisensor-Based Human Detection and Tracking for Mobile Service Robots, IEEE Trans. SMC 2009 39.

[12] Martins, M, Frizera, A, Ceres, R. & Santos, C. Legs tracking for walker-rehabilitation purposes, ICBRB 2014.

[13] Zhao, X, Zhu, Z, Liu, M, Zhao, C. et al. A Smart Robotic Walker With Intelligent Close-Proximity Interaction Capabilities for Elderly Mobility Safety, Front Neurorobot. 2020.

[14] Taheri, O, Salarieh, H. & Alasty, A. Human Leg Motion Tracking by Fusing IMUs and RGB Camera Data Using Extended Kalman Filter, arXiv:2011.00574.

[15] Gavrilova, M, Ahmed, F, Azam, S, Paul, P. et al. Emerging Trends in Security System Design Using the Concept of Social Behavioural Biometrics, Information Fusion for Cyber-Security Analytics (2017).

[16] Arras, K, Lau, B Grzonka, S, Luber, M, Mozos, O, Meyer-Delius, D. & Burgard, W. Range-Based People Detection and Tracking for Socially Enabled Service Robots, Towards Service Robots for Everyday Environments (2012).

[17] Leigh, A, Pineau, J, Olmedo, N. & Zhang, H. Person tracking and following with 2D laser scanners, ICRA 2015.

[18] Beyer, L, Hermans, A, Linder, T, Arras, K. & Leibe, B. Deep Person Detection in 2D Range Data, CVPR 2018.

[19] Beyer, L, Hermans, A. & Leibe, B. DROW: Real-Time Deep Learning-Based Wheelchair Detection in 2-D Range Data, RAL 2016.

[20] Guerrero-Higueras, Á, Álvarez-Aparicio, C, Olivera, M. C. et al. Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments, Front Neurorobot. 2019.

[21] Ronneberger, O, Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015.

[22] Chalvatzaki, G, Papageorgiou, X, Tzafestas, C. & Maragos, P. Augmented Human State Estimation Using Interacting Multiple Model Particle Filters With Probabilistic Data Association, RAL 2018.

[23] Redmon, J, Divvala, S, Girshick R & Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016.

[24] Lu, Y, Lu, C. & Tang, C. Online Video Object Detection Using Association LSTM, ICCV 2017.

[25] Ning, G. et al. Spatially supervised recurrent convolutional neural networks for visual object tracking, ISCAS 2017.

[26] Cheema, N, Hosseini, S, Sprenger, J. et al. Dilated Temporal Fully-Convolutional Network for Semantic Segmentation of Motion Capture Data, SCA 2018.

[27] Chalvatzaki, G, Koutras, P, Hadfield, J, Papageorgiou, X, Tzafestas, C, & Maragos, P. LSTM-based network for human gait stability prediction in an intelligent robotic walker, ICRA 2019..

[28] Perry, J. & Davids, J. Gait analysis: normal and pathological function, Journal of Pediatric Orthopaedics 1992 12(6).

[29] Hochreiter, S. & Schmidhuber, J. Long Short-term Memory, Neural Computation 1997 9(8).