Detecting unsafe use of a four-legged walker using IoT and deep learning

Siavash Khodadadeh*, Sharare Zehtabian*, Jonathan Guilbe*, Ross Pearlman‡, Bradley J. Willenberg‡,
Brian Kim†, Edward A. Ross‡, Ladislau Bölöni* and Damla Turgut*
*Department of Computer Science, University of Central Florida,
{siavash.khodadadeh, sharare.zehtabian, jguilbe}@knights.ucf.edu,{lboloni, turgut}@cs.ucf.edu
†Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA
Brian.Kim@ucf.edu
‡Department of Internal Medicine, University of Central Florida, Orlando, FL, USA
Ross.Pearlman@knights.ucf.edu, {Bradley.Willenberg, Edward.Ross}@ucf.edu

***Abstract*—Four legged walkers are used by many elderly persons to retain mobility. They are also used by patients recovering from leg injuries to facilitate rehabilitation. Unfortunately, these walkers are also associated with many injuries, some of which are caused by incorrect use. In this paper, we describe a walker augmented with IoT sensors which continuously monitors the weight distribution on the legs of the walker. We describe an approach where this data stream is processed by a deep neural network based classifier, which learns to recognize dangerous use patterns that can lead to falls and injury. The classifier is trained by providing examples of unsafe use, thus eliminating the costly engineering necessary to customize the algorithm to the specific user and walker. By alerting the user in real time about unsafe use patterns, the user can learn the correct and safe use of the walker.**

I. Introduction

Four legged walkers are used by millions of people worldwide both as a way for the elderly to retain mobility as well as a rehabilitation tool for re-acquiring mobility after a trauma or an accident. On the other hand, the use of walkers is also associated with many injuries, falls and broken limbs. There can be many reasons for such injuries. Patients might take risks relying on the walker which they would not normally take. People might also be distracted by the use of the walker, paying less attention to their surroundings. While a walker reduces the weight load on the legs and lower body, it increases the load on the hands and the upper body. Although there is extensive literature on this subject [1], there is no consensus on the relative weight of these factors. What is clear, however, is that a significant fraction of accidents is associated with the incorrect use of the walker.

We developed an IoT-augmented walker that continuously collects sensor information from four weight sensors mounted under the legs and a proximity sensor that measures the distance of the user from the walker. The IoT augmented walker can be used in multiple application scenarios - it can collect data about the patient's lifestyle, it can track compliance with the prescription in the case of rehabilitation, and it can also act as a real-time autonomous agent constantly monitoring the patient's use of the walker and providing feedback.

Considering the significant danger of falls and injuries associated with the use of a walker, one of the most important features of such a device is to train users in the correct use of the walker and to alarm them to possible unsafe use patterns. Thus, the user can acquire skills that he or she can deploy also on walkers without IoT augmentation.

One of the challenges in this task is the specification of what constitutes unsafe use for a particular user and walker. While there has been some research concerning the static and dynamic equilibrium of a person using a walker or a cane [2], the proposed models are too complex to be applicable in practice to the calculation of whether certain usage patterns are safe or dangerous. Beyond the difficulty of the physics involved, a major problem in developing such a model is the fact that it depends on the ability of the person to control his or her limbs – a gait that is safe for a healthy person might be unsafe for a person that has balance problems after recovering from a stroke.

In this paper we propose an approach where a deep neural network learns to recognize unsafe usage patterns of the walker, as labeled by the expert (such as a physical therapist). During use, the resulting system can recognize these usage patterns in real time, alerting the patient to dangerous situations. This feedback can then be used by the patient to learn the correct use of the walker and avoid dangerous situations.

II. Related Work

Four legged walkers can be used to promote activity and improve balance and mobility of elderly people [1], [3], in the recovery phase for people after stroke or lower limb injury, or to protect patients with balance problems from falling [4]. Furthermore, developing systems to help physical therapists make more accurate and appropriate clinical decisions quickly is effective in response to the needs of detailed health information during rehabilitation [5], [6] [7]. Thus, walkers need to be safe, comfortable, and easy to use.

The popularity of walkers opens the opportunity to add technological augmentations that can be used to monitor a number of parameters of the users' health condition. This resulted in the development of several *smart walker* projects [8], [9], [10], [11], [12], [13], [14], [15]. Most smart walkers measure parameters such as weight distribution, posture and gait of the user. Some smart walkers also capture physiological data, or record 2D or 3D images of the patient and the environment. Many recent walker projects take advantage of client-server architectures that allow the storage and post-processing of the data in the cloud. Some smart walker projects also provide actuation capabilities, representing a bridge towards assistive robotics and exoskeleton technology. An exhaustive survey of the previous smart walker projects is beyond the scope of this paper. In the following, we briefly review several projects illustrating the diversity of projects.

Spenko et al. [16] designed a smart walker that can record the user's activity level by using an ECG-based pulse monitor. The system is able to calculate speed of the user and stride-to-stride variability, and also gait asymmetry. The smart walker can predict the likelihood of falls by using variability and gait asymmetry. However, the complexity of the system and the type of the data collected restricts the use of this system to scenarios where physicians are present.

Hirata et al. [17] proposed RT Walker, a passive-type intelligent walker which can distinguish walking, stop, and emergency states by utilizing a laser range finder. RT Walker can estimate the center of gravity of the user and compare it to the normal distribution during walking.

Valadão et al. [18] developed a smart walker without any attached sensors. By using a laser range sensor, the smart walker is able to calculate the speed, position, and orientation of the user inferred from the distance between the walker and the user's legs. The situations when laser sensor just detects one leg implies the user might fall. There are other safety rules for this walker which are high speed, backward movements, obstacles, and counter exceeded limit that prompt the walker's safety supervisor to take actions to ensure the user's safety.

Feltner et al. [19] developed a smart walker for visually impaired individuals. They used the depth images captured by Kinect RGB-D camera to predict the distance of the obstacles to the user. They leveraged this information to detect obstacles that may endanger the user.

The walkers augmented with laser range sensors are wheeled walkers. Wheeled walkers are useful because the user does not need to pick them up for each step and can easily move around with it. On the other hand, four legged walkers are considered the most stable of the conventional walkers and can be used for people who do not have good balance [20].

III. Hardware and software implementation of the IoT augmented walker

The IoT-augmented walker starts with an off-the-shelf four-legged walker that has been augmented with sensors, computing devices and user interface components (see Figure 1). Other than a slight increase in weight, there is no change in the functionality of the walker.

The weight distribution applied to the four legs of the walker are captured by four load cells, each connected to a HX711 amplifier. The proximity of the user to the sensor is measured using the VL53L0X time-of-flight laser ranging module. Finally, orientation data is provided by the Adafruit BNO005 absolute orientation sensor which combines a MEMS accelerometer, magnetometer and gyroscope on a single die with an ARM Cortex-M0 processor. We used these sensors as they are the most common for fall prevention research [10].

The sensor data is collected by a Raspberry Pi-based main computer housed in custom 3D-printed housing. The top of the box serves as a display that provides feedback to the patient using the device. The computer can connect to available WiFi networks for real-time data upload (this is not necessary for normal operation). The physical therapist can configure the device either through WiFi or through a keyboard attached to the USB port. The patient does not need to perform any other interaction with the device except turning it on and off, and recharging the battery with a standard charger. The internal battery has 10000 mAh capacity and according to our testing, the device can operate 12+ hours while the software is running.

The software running on the IoT-augmented walker must perform several distinct functions. In the background, it initializes the sensors, and continuously collects data from them, performs initial filtering, stores and, if required, uploads it to a centralized web service. The web service is run on a dedicated server or the cloud and is implemented using the Django Python-based web framework [21]. This functionality is performed in the background and does not require user interaction.

The user interacts with the IoT-augmented walker through a simple visual user interface that provides continuous real-time feedback to the user about whether it is using the walker correctly and whether it follows the prescription.

IV. Preprocessing the data stream

The stream of data as captured from the sensors on the walker is extremely noisy, and in its raw form is not suitable to be presented to the user or used as an input to higher level algorithms. In this section we discuss the preprocessing we apply to the raw data flow. The two steps of the preprocessing are (a) filtering the sensor data for noise and (b) segmenting the data stream into the individual steps taken by the patient.

The datastream collected by the load sensors contains noise coming from multiple sources: the inherent noise of the sensor, mechanical vibrations and deformations within the material of the walker, imperfect contact between the feet of the walker and the ground, and the changing load and possible tremor of the user's hand transmitted to the walker. We use a Kalman filter to eliminate the Gaussian components of the noise which occur primarily from sensor noise and vibrations. As Figure 2-top shows, this makes the data significantly cleaner, but does not completely eliminate the noise coming from the user's hand trying to adjust to the correct load distribution.

The second preprocessing step is the temporal segmentation of the four independent data streams into steps. This is a non-trivial challenge due to the asymmetric load of the legs, the variability in the step sizes and noisy data. We define the beginning of a step as the moment when the walker is completely lifted up from the ground and the amount of pressure

Fig. 1. The IoT augmented walker from the point of view of the user (left), and a view of the available connections (right)

applied to the leg is in a local minimum. Due to the noise and the temporal differences between the four data channels, there are many local minima in the moment of transitioning between the steps. The approach we take is based on applying a wavelet convolution to the data with a range of wavelets of different widths (1-20). We choose the minima that appears in the most length scales with a sufficiently high signal to noise ratio. The resulting segmentation is shown in Figure 2-bottom.

V. A NEURAL NETWORK CLASSIFIER FOR UNSAFE WALKER USE PATTERNS

As we discussed in the introduction, we lack the formal models to identify unsafe use patterns from physical principles. Another possibility is building a model using knowledge engineering – we would interview an expert such as a physical therapist and build a model that captures the expert's knowledge. Unfortunately, it is difficult to express expert physiotherapy knowledge in terms of four streams of noisy sensor data.

The approach we take is to train a deep neural network classifier using supervised learning to distinguish between safe and unsafe patterns of use. Supervised learning requires labeled training data. As the four streams of sensor data are not human comprehensible, in our data acquisition process we rely on the synchronized recording of video and sensor readings. During the *data acquisition phase* an experimenter, under the instruction of an expert enacts both safe and unsafe modes of utilization. A video of the experimenter and the sensor readings are recorded with shared timestamps. At the *labeling phase* the expert watches the video (with the possibility of slowdown and rewind) and provides a label for the time slots in which the walker was incorrectly used. These labels are then applied to the sensor readings collected at the same timeslots.

We expect unsafe use patterns to happen with a lower rate in comparison with the safe use patterns because normally subjects try to avoid unsafe actions. We collected data in five different sets of experiments to be able to create a balanced dataset. The first set corresponds to correct and safe usage in which the user is asked to step correctly with the walker. They are explicitly told to be very careful to put all four legs of the walker together on the ground. The next four set of experiments correspond to unsafe use patterns. In the second set of experiments, the subject is asked to put the rear legs first and then the front legs. In the third set of experiments, we collect data while asking the subjects to put front legs of the walker on the ground first followed by rear legs. In the fourth set of experiments, the subjects are asked to put two left legs sooner than two right legs and vice versa for the fifth set of experiments. In all of these experiments, the path which users follow is the same which is a narrow round path around a table.

For each experiment, we collect sensors data as well as video recordings. The expert then looks at the video and selects the intervals which correspond to unsafe steps. Based on this supervised information, we label the corresponding sensors' reading data for that interval. We want the intervals which are longer than 2.5 seconds, and we ignore the interval if its length is shorter than 2.5 seconds.

During training, we have to sample data from all of these experiments (i.e. four sets of unsafe patterns and one set of safe patterns). We sample N intervals from each of the unsafe patterns sets and $4 \times N$ intervals from the safe patterns set.

The input data to the classifier must have a temporal component because unsafe use patterns, which usually deal with conditions of dynamic equilibrium, cannot be identified from single snapshots of sensor data. Thus, our input data will be a window of 25 recordings (lines of data) from the four sensor streams, creating an input vector of size 100. With a sampling rate of 10Hz (because we store 10 recordings each second), this corresponds to a 2.5 second interval for determining an unsafe pattern. An alternative to this sliding window approach would be the use of a classifier with memory, such as an LSTM recurrent neural network, a possible choice that is beyond the scope of this paper.

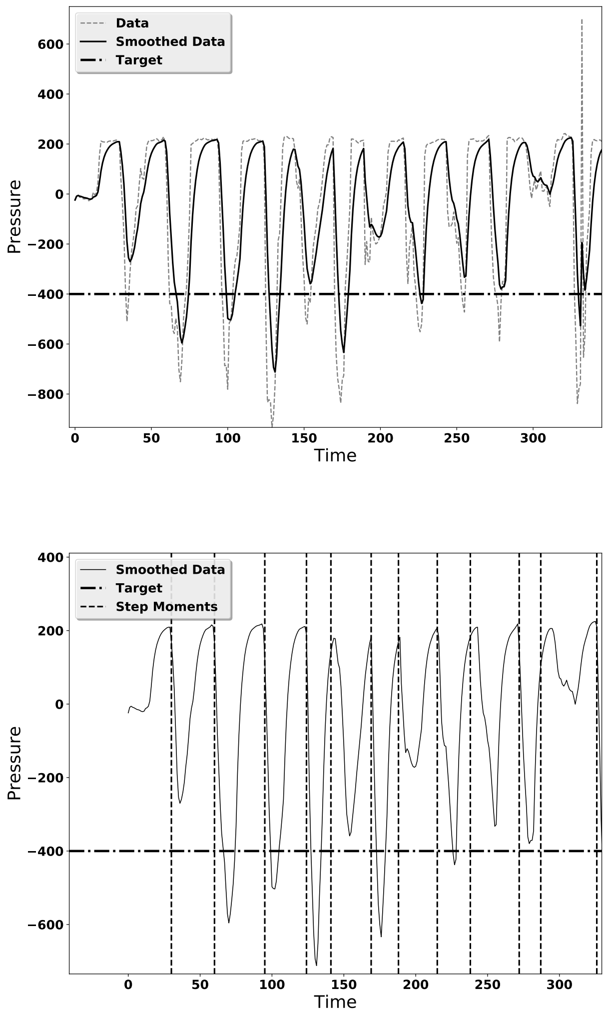

Fig. 2. Preprocessing the raw data steam. Top: filtering the input signal using a Kalman filter. Bottom: segmenting the steps taken by the user with a wavelet transformation. In both figures, the target line shows the target load - this is the value to which the minimums should be aligned if the walker is used according to prescription.

The classifier is implemented as a feed forward fully connected neural network with three hidden layers of 100, 50 and 50 neurons respectively and a ReLU activation function. For better generalization, dropout layers are added between the fully connected layers with a keep probability of 0.5. We also use batch normalization in order to make the training phase faster [22]. During the training we update moving average and moving variance with respect to each mini-batch and keep track of them for evaluation. We also use both centering and scaling. The output layer has two neurons corresponding to recognized safe and unsafe use patterns. The network was trained using the Adam optimizer [23] and the cross entropy loss defined as

$$\mathcal{L}(y, \hat{y}) = -\mathbb{E}\left[y \cdot log(\hat{y})\right] \tag{1}$$

where y is the actual label and $\hat{y}$ is the probability vector of each class generated by the network.

Note that even with the windowing technique, the classifier provides a classification result every 0.1 seconds (every 0.1 seconds, we look at the last 25 records), a rate much higher compared to the needs of the application as training feedback. In early experiments, we also noticed that at this high data rate we could not avoid the presence of occasional false positives and false negatives in the data stream. Thus, we took advantage of the high rate of data to perform additional processing on the output stream of the classifier by applying a technique of mathematical morphology, the $n = 5$ steps binary erosion followed by the same number of steps of binary dilation of the output signal treated as a temporal signal. This technique is often used in computer vision for recognizing image shapes – in this case we are using it in the temporal domain. If P is our prediction signal, our processed prediction $\hat{P}$ will be

$$\hat{P} = ((P \ominus M) \oplus M) \quad \text{where} \quad M = [1, 1, 1]. \tag{2}$$

VI. Experiments

A. Data collection

We collected real world data from the IoT-augmented walker for training and validation purposes. Six different healthy subjects (two women and four men, height from $5'2''$ to $5'11''$, and weight from $93lb$ to $178lb$) were instructed to use the walker both in a safe and unsafe use patterns. Unsafe use patterns included steps that are too long, highly asymmetric load on the walker's left or right side, dangling legs, and putting the rear legs of the walker down first then gradually lowering the front ones and vice versa. During the experiments, the walker recorded measurements in a plain text logfile in the format: {`timestamp`, `w_left_forward`, `w_right_forward`, `w_left_rear`, `w_right_rear`}, with the measurements being recorded at a frequency of 10Hz. The values were filtered and the steps separated using the algorithms mentioned in Section IV. Additionally, we recorded a manual step segmentation from an assistant who observed the process. The experiments were also recorded on a video stream synchronized to the timestamps on the walker. This video stream was later used for the external labeling of the recorded data, which added a boolean variable labeling the current recording as safe or unsafe to the logs.

We collected 12398 data points. The data points were split into training, validation and test data using a 80/10/10 ratio. Every subject was asked to perform 6 experiments. The first experiment corresponds to correct use of the walker. In the next five experiments, the subjects were asked to perform one of the unsafe patterns while walking. To ensure the desired distribution of the safe and unsafe use patterns, the number of steps sampled from each of these experiments were balanced such that the number of safe steps are equal to the number of unsafe steps.

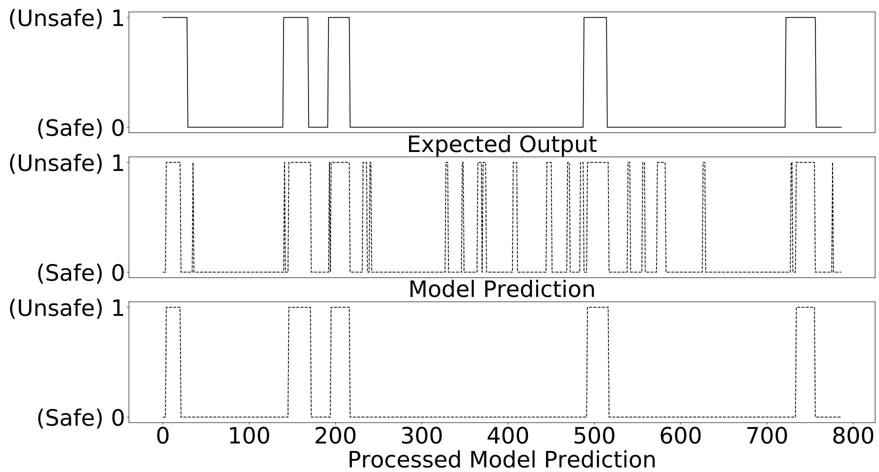

Fig. 3. The final output of the unsafe behavior detection on a continuous data stream. Top: The safe/unsafe labels provided by the expert annotator. Middle: The classifier output – note the existence of both false positives and false negatives. Bottom: Classifier output processed with repeated binary erosion and dilation.

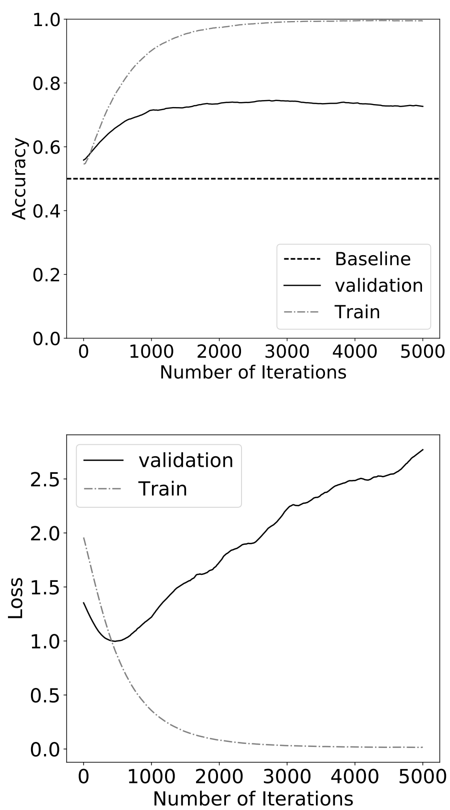

Fig. 4. The training process of the classifier. Top: Training and validation accuracy of the model with respect to the number of iterations. The diagram also includes a baseline classifier that classifies every sample as safe. Bottom: Evolution of the loss function with respect to the number of iterations.

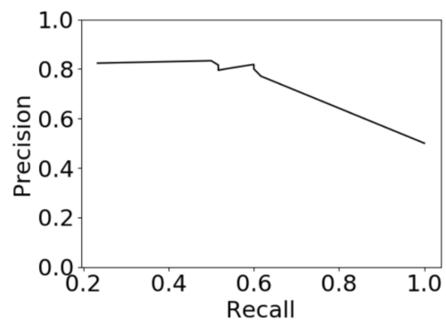

Fig. 5. Precision-Recall curve for the classifier.

B. Training the classifier

The classifier described in Section V was implemented in TensorFlow 1.7, and trained using the training data obtained as above. As the network needed to capture the different unsafe pattern modalities, its large number of parameters were also prone to overfitting. We used the technique of early stopping [24] and tracked the evolution of the validation error and loss through the training process. Figure 4 shows the classification accuracy (top) and the loss function (bottom) on the training and the validation data set. We can see that after approximately 700 iterations (with a mini-batch size of 64), the training loss continues to decrease, but the validation loss has an upward trend. We saved snapshots of the neural network, and retained the network at the point when the validation loss started to trend upward.

C. Evaluation

The precision-recall curve of the trained classifier is shown in Figure 5. As expected, the classifier is working, but it is not particularly accurate. The reason for this is that the input is noisy, but also the fact that the temporal labeling of an unsafe movement pattern is somewhat arbitrary, and it might not exactly match the beginning and the end of the unsafe behavior. Thus, our quality criteria is not whether we can identify safe/unsafe behavior at a 10Hz frequency, but whether we can flag steps where unsafe behavior occurs and avoid flagging steps which has been executed safely.

As we discussed in Section V, the temporal output of the classifier is further processed with repeated steps of binary erosion and dilation. The result of this process is illustrated in Figure 3. The top graph shows the real labels assigned by the experimenter. The middle graph shows the output of the classifier – as expected, it shows both false positives and false negatives. Finally, the bottom graph shows the classifier results after the repeated steps of binary erosion and dilation. We note that the bottom graph correctly identifies all the steps that included unsafe behavior and does not classify any of the safe steps as unsafe, although it does not always match the time point where the transition from safe to unsafe and the reverse happens in the original labels.

VII. Conclusions

In this paper, we described an approach that allows us to detect unsafe use patterns in a four legged walker based on sensor data collected from an IoT-augmented walker. We trained a deep neural network based classifier over sliding windows of the sensor streams. By feeding the classifier output into a mathematical morphology based postprocessing unit, we were able to identify unsafe steps with a high confidence. This data can be used as a feedback mechanism for user training in a clinic and more importantly elsewhere when the user uses the walker alone without the presence of the physical therapist.

Future work includes validating the work with elderly or disabled patients, determining whether the training model is valid across patients and extending the ability to identify different patterns of use. Furthermore, instead of alarming for incorrect usage, we can provide feedback about how to correctly use the walker.

Acknowledgement

This work is partially supported by the National Science Foundation REU program under Award No. 1560302 and by the Department of Internal Medicine at the UCF College of Medicine. Any opinions, findings, and conclusions and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] H. Bateni and B. E. Maki, "Assistive devices for balance and mobility: benefits, demands, and adverse consequences," *Archives of physical medicine and rehabilitation*, vol. 86, no. 1, pp. 134–145, 2005.

[2] Y. Tagawa, N. Shiba, S. Matsuo, and T. Yamashita, "Analysis of human abnormal walking using a multi-body model: joint models for abnormal walking and walking aids to reduce compensatory action," *Journal of Biomechanics*, vol. 33, no. 11, pp. 1405–1414, 2000.

[3] A.-L. Salminen, Å. Brandt, K. Samuelsson, O. Töytäri, and A. Malmivaara, "Mobility devices to promote activity and participation: a systematic review," *Journal of rehabilitation medicine*, vol. 41, no. 9, pp. 697–706, 2009.

[4] W. Graafmans, P. Lips, G. Wijlhuizen, S. Pluijm, and L. Bouter, "Daily physical activity and the use of a walking aid in relation to falls in elderly people in a residential care setting," *Zeitschrift für Gerontologie und Geriatrie*, vol. 36, no. 1, pp. 23–28, 2003.

[5] I. Edwards, M. Jones, J. Carr, A. Braunack-Mayer, and G. M. Jensen, "Clinical reasoning strategies in physical therapy," *Physical therapy*, vol. 84, no. 4, pp. 312–330, 2004.

[6] P. M. Masley, C.-L. Havrilko, M. R. Mahnensmith, M. Aubert, and D. U. Jette, "Physical therapist practice in the acute care setting: a qualitative study," *Physical therapy*, vol. 91, no. 6, pp. 906–919, 2011.

[7] B. A. Smith, C. J. Fields, and N. Fernandez, "Physical therapists make accurate and appropriate discharge recommendations for patients who are acutely ill," *Physical therapy*, vol. 90, no. 5, pp. 693–703, 2010.

[8] O. Postolache, J. M. D. Pereira, V. Viegas, L. Pedro, P. S. Girão, R. Oliveira, and G. Postolache, "Smart walker solutions for physical rehabilitation," *IEEE Instrumentation & Measurement Magazine*, vol. 18, no. 5, pp. 21–30, 2015.

[9] S. Zehtabian, S. Khodadadeh, R. Pearlman, B. Willenberg, B. Kim, D. Turgut, L. Bölöni, and E. A. Ross, "Supporting rehabilitation prescription compliance with an IoT-augmented four-legged walker," in *2nd Workshop on AI for Aging, Rehabilitation and Independent Assisted Living (ARIAL'18)*, July 2018.

[10] B. J. Willenberg, S. Pal, L. Khan, C. Cepeda, R. Pearlman, W. Perez, T. J. Tomlinson, M. Pita, P. Pabian, A. Golden, and E. A. Ross, "A smart-walker system for fall prevention and rehabilitation," in *Annual Meeting of the Bio-Medical Engineering Society*, October 2016.

[11] O. Chuy, Y. Hirata, Z. Wang, and K. Kosuge, "Motion control algorithms for a new intelligent robotic walker in emulating ambulatory device function," in *Proc. of IEEE Int'l Conf. on Mechatronics and Automation*, vol. 3, 2005, pp. 1509–1514.

[12] O. Chuy, Y. Hirata, and K. Kosuge, "Control of walking support system based on variable center of rotation," in *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, (IROS 2004)*, vol. 3, 2004, pp. 2289–2294.

[13] S. Dubowsky, F. Genot, S. Godding, H. Kozono, A. Skwersky, H. Yu, and L. S. Yu, "PAMM - a robotic aid to the elderly for mobility assistance and monitoring: a "helping-hand" for the elderly," in *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA-00)*, vol. 1, 2000, pp. 570–576.

[14] G. Lacey and K. M. Dawson-Howe, "The application of robotics to a mobility aid for the elderly blind," *Robotics and Autonomous Systems*, vol. 23, no. 4, pp. 245–252, 1998.

[15] R. Schraft, C. Schaeffer, and T. May, "Care-O-bot/sup TM/: the concept of a system for assisting elderly or disabled persons in home environments," in *Proc. of the 24th Annual Conf. of the IEEE in Industrial Electronics Society, IECON'98.*, vol. 4, 1998, pp. 2476–2481.

[16] 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.

[17] Y. Hirata, A. Muraki, and K. Kosuge, "Motion control of intelligent passive-type walker for fall-prevention function based on estimation of user state," in *Proc. of IEEE Int'l Conf. on Robotics and Automation, ICRA*, 2006, pp. 3498–3503.

[18] C. Valadão, E. Caldeira, T. Bastos-Filho, A. Frizera-Neto, and R. Carelli, "A new controller for a smart walker based on human-robot formation," *Sensors*, vol. 16, no. 7, p. 1116, 2016.

[19] C. Feltner, J. Guilbe, S. Zehtabian, S. Khodadadeh, L. Bölöni, and D. Turgut, "Smart walker for the visually impaired," in *Proc. of IEEE Int'l Conf. on Communication (ICC)*, May 2019.

[20] G. Lacey, S. Mac Namara, and K. M. Dawson-Howe, "Personal adaptive mobility aid for the infirm and elderly blind," in *Assistive Technology and Artificial Intelligence*. Springer, 1998, pp. 211–220.

[21] D. S. Foundation, "Django," https://djangoproject.com.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.