International Journal of Semantic Computing
Vol. 13, No. 2 (2019) 245–257

DOI: 10.1142/S1793351X19400117

Gait Pattern Recognition Using a Smartwatch Assisting Postoperative Physiotherapy

Athanasios I. Kyritsis

Information Science Institute, GSEM/CUI
University of Geneva, Battelle A, Route de Drize 7
1227 Carouge, Switzerland
athanasios.kyritsis@unige.ch

Geoffrey Willems

Physiotherapy Center, Hirslanden Clinique La Colline
Avenue de la Roseraie 76 A
1205 Geneva, Switzerland
geoffrey.willems@lacolline.ch

Michel Deriaz and Dimitri Konstantas

Information Science Institute, GSEM/CUI
University of Geneva, Battelle A, Route de Drize 7
1227 Carouge, Switzerland
michel.deriaz@unige.ch
dimitri.konstantas@unige.ch

Postoperative rehabilitation is led by physiotherapists and is a vital program that re-establishes joint motion and strengthens the muscles around the joint after an orthopedic surgery. Modern smart devices have affected every aspect of human life. Newly developed technologies have disrupted the way various industries operate, including the healthcare one. Extensive research has been carried out on how smartphone inertial sensors can be used for activity recognition. However, there are very few studies on systems that monitor patients and detect different gait patterns in order to assist the work of physiotherapists during the said rehabilitation phase, even outside the time-limited physiotherapy sessions. In this paper, we are presenting a gait recognition system that was developed to detect different gait patterns. The proposed system was trained, tested and validated with data of people who have undergone lower body orthopedic surgery, recorded by Hirslanden Clinique La Colline, an orthopedic clinic in Geneva, Switzerland. Nine different gait classes were labeled by professional physiotherapists. After extracting both time and frequency domain features from the time series data, several machine learning models were tested including a fully connected neural network. Raw time series data were also fed into a convolutional neural network.

Keywords: Activity recognition; feature extraction; machine learning; pattern recognition; smart devices; wearable computers; wearable sensors.

1. Introduction and Related Work

The role of physiotherapy following an orthopedic surgery is to assist the patient return to normal activities of daily living. Doctors and physiotherapists help the patient achieve this by prescribing suitable exercises that will establish the rehabilitation goals. There is a significant body of evidence coming from systematic reviews and controlled trials that dictate the best practices in physiotherapy [1]. Proper evaluation guarantees the effectiveness of physiotherapy [2] for a wide variety of medical conditions, including recovering after a lower body orthopedic operation.

Gait refers to a person's manner of walking and is influenced by age, personality, mood and sociocultural factors [3]. Several reasons including a lower body operation may lead to either temporary or permanent gait disorders. Any such disorder is typically thoroughly investigated by the physiotherapist who then suggests a specific treatment to the patient. There are various tools at the disposal of the physiotherapists, and many robotic solutions are being created in order to help people walk or to act as an aid during a physiotherapy session [4]. These robot-assisted gait solutions may be used as an excellent companion to conventional therapy and improve the independence and the gait capacity of the patient [5].

Activity recognition (AR) has emerged as a key research domain in computer science. The approaches for AR can be roughly divided into two categories: the camera-based ones [6], where gestures and activities are inferred from still images or videos using computer vision techniques, and the inertial sensor-based ones, where one or more body-worn sensors are used [7]. Any AR system includes many variables such as the definition of the classes of interest, the experiment design, the sensors, the data handling procedure, and the performance evaluation. These variable components can be implemented in a variety of ways [8] having an impact on the overall performance of the system.

The increased availability of inertial sensors due to the omnipresence of smartphones and particularly smartwatches has enabled AR to become an essential context-awareness tool for mobile and ubiquitous computing. Sensors in modern consumer electronics provide reasonably accurate recordings when compared to research monitors [9]. This is why these devices prove to have clinical utility, although they continue to be underutilized in the healthcare industry [10].

Besides recognizing daily activities, inertial sensors have been used in gait pattern analysis. In most studies accelerometers are attached to the legs or feet, but gait patterns can be also extracted from data recorded from sensors attached to the upper body [11]. Common smartphone accelerometers have been used to detect different gait events [12]. In a similar manner, smartwatches that contain inertial sensors can be used for gait recognition. Unlike smartphones, smartwatches tend to be worn in the same location and the same orientation and can be even used for gait-based biometrics based on the accelerometer and the gyroscope data [13].

Various recovery programs have been developed to improve the recovery time after surgery [14]. Wireless monitoring of mobility after a major operation has the potential of improving services provided by healthcare professionals [15]. With the proposed system, we incorporate smartwatches into the routine care of patients who have undergone a lower body operation in order to monitor their gait patterns. Doing so will enhance the patient–physiotherapist relationship, respect the patients' autonomy regarding their healthcare and provide a remote monitoring solution to the physiotherapist in charge.

The rest of the paper is organized as follows. In Sec. 2, we discuss the system that we have developed. We present the data acquisition tools and the data preprocessing step. In Sec. 3, we present the experiment that we have conducted and we evaluate the performance of the overall system by training machine learning models and neural networks. Finally, we conclude our work in Sec. 4.

2. System Overview

2.1. *Gait classification*

The physiotherapists of Hirslanden Clinique La Colline, an orthopedic clinic in Geneva, Switzerland, compiled a list of the gait patterns of interest to our system. The patterns include walking with crutches with various levels of weight-bearing, walking with different frames, limping and walking normally. Table 1 includes the list of all the 9 gait patterns that our system should detect.

Table 1. Classification of gait patterns for recognition.

Category	Class
No aid	Limping
	Walking
Crutches	Unladen
	Rolled out
	Laden 10 kg
	Laden 20 kg
	According to pain
Frame	Without wheels
	With wheels

2.2. *Workflow*

The developed system comprises three components, the smartwatch, the smartphone, and the web server. Figure 1 presents the flow of data in the proposed system. The system is meant to be used during the rehabilitation phase, that is the time that the patient is undergoing physiotherapy after a lower body surgery. During physiotherapy sessions in the clinic, any patient is walking while wearing a smartwatch that tracks wrist movements. At the same time, the physiotherapist is labeling on a smartphone any physiotherapy session with the observed gait pattern of the patient. All these data from multiple patients and physiotherapy sessions are uploaded to the web server, where a user-independent machine learning model is trained.

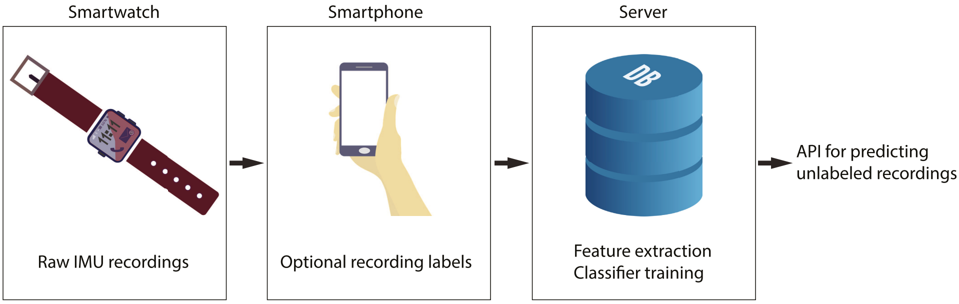

Fig. 1. Summary of the implementation of the gait recognition system.

During everyday life, through the rehabilitation phase, the patient is wearing a given smartwatch. Throughout the day, the smartwatch is passively recording gait sessions of unknown gait patterns when the patient is moving. These recordings are uploaded from the smartwatch to the web server. Using the trained machine learning model, those new recordings are classified into the predefined gait patterns. Using the web server, the physiotherapists can monitor how each patient's gait pattern is evolving, even between physiotherapy sessions.

2.3. *System implementation*

Wrist movements of the patients are recorded using the three-axis accelerometer and the three-axis gyroscope of an Android smartwatch running Wear OS. The accelerometer sensor provides a three-dimensional vector representing acceleration along each device axis, excluding gravity. The gyroscope sensor measures the angular velocity of each axis of the device. Recordings can be made either on-demand during a physiotherapy session when the physiotherapist can provide the ground truth with the observed gait pattern, or by transparently monitoring the movement of the user throughout the day and saving only sessions where prolonged movement or steps are identified.

At the end of every on-demand recording, sensor data are sent from the smartwatch to the connected Android smartphone. The smartphone is used by physiotherapists to label each recording during a physiotherapy session with the identified gait pattern. The recordings that are produced during the monitoring phase of the system during the whole rehabilitation program, naturally have no ground truth label and are directly sent from the patient's smartwatch to the web server.

Every recording is saved to the web server. On every upload of a new recording, the web server is extracting the features that will be later used for machine learning. Training of the selected user-independent machine learning classifier is run periodically when enough new labeled recordings from multiple users have been obtained. On the other hand, the server exposes an API with which the unknown gait patterns of the nonlabeled recordings can be predicted. The physiotherapist can query the server in order to monitor what is the dominant detected gait pattern of a specific time and how it evolves during the rehabilitation program.

2.4. *Data preprocessing*

The accelerometer and the gyroscope sensors of the smartwatch that we have used did not provide a constant sampling rate throughout the recordings. This is why the raw sensor data were resampled with a constant sampling frequency of 60 Hz. This frequency was selected for this study as it is higher than the 20 Hz commonly required to assess daily living [16] and also lower than what typical off-the-shelf inertial measurement unit components can achieve. Features forming the feature vector used for machine learning were derived from these time series data and these were the raw data fed to the convolutional neural network.

3. Experiment and Evaluation

Physiotherapists of the Hirslanden Clinique La Colline recorded wrist movements of patients walking soon after they have undergone a lower body orthopedic surgery. During all recordings, the physiotherapist was in close proximity to the patient, in order to guarantee the correct ground truth annotation and the cleanliness of the data. In total, 48 recordings from 33 different patients were made over a period of 4 months. The recordings amount to a total time of 155 min of labeled gait patterns.

3.1. *Feature engineering*

Both time and frequency domain features were computed for both sensors over a selected time window. The time domain features include the mean, the standard deviation, the median, the skewness, the kurtosis, the 25th and the 75th percentile, and the squared sum of the components under the 25th and the 75th percentile. Those were derived from the resultant vector computed by the three, x, y and z, components that each sensor provides.

For the frequency domain features, a Fast Fourier Transform (FFT) was performed after normalization on the windows, and the features were computed per axis. Those features include the maximum frequency, the sum of heights of frequency components below 5 Hz and the number of peaks in the spectrum below 5 Hz, as it was noticed that most of the signal strength lied between 0–5 Hz. The selection of the features was based on a feature importance analysis presented in a previous work of ours [8]. All the features extracted for this study are summarized in Table 2.

3.2. *Machine learning*

The classifiers that we have evaluated are Light Gradient Boosting Machine (LGBM) [17], Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), Decision Tree (DT), Extra Trees (ET) and k-Nearest Neighbors (kNN). Each recording is segmented into multiple time windows. The features were

Table 2. Extracted features per sensor used in machine learning.

Domain	Features	No of features
Time (resultant vector)	Mean Standard deviation Median Skewness Kurtosis 25th percentile 75th percentile Sq. sum of $<$ 25th perc. Sq. sum of $<$ 75th perc.	9
Frequency (per axis)	Maximum frequency Sum of 5 Hz Number of peaks	9

computed over a time window of five seconds with a step size of one second, so there was a four-second overlap between consecutive windows. This value for the time window was identified in a previous work of ours [8] as a good candidate since it is large enough to contain useful information regarding the activity and small enough to increase the number of the produced segments during segmentation. The segmentation of any given recording is depicted in Fig. 2. The constructed dataset contained in total 9089 observations.

Unfortunately, the acquired dataset was imbalanced. The reasons were either lack of availability of patients with a gait pattern belonging to one of the minority classes

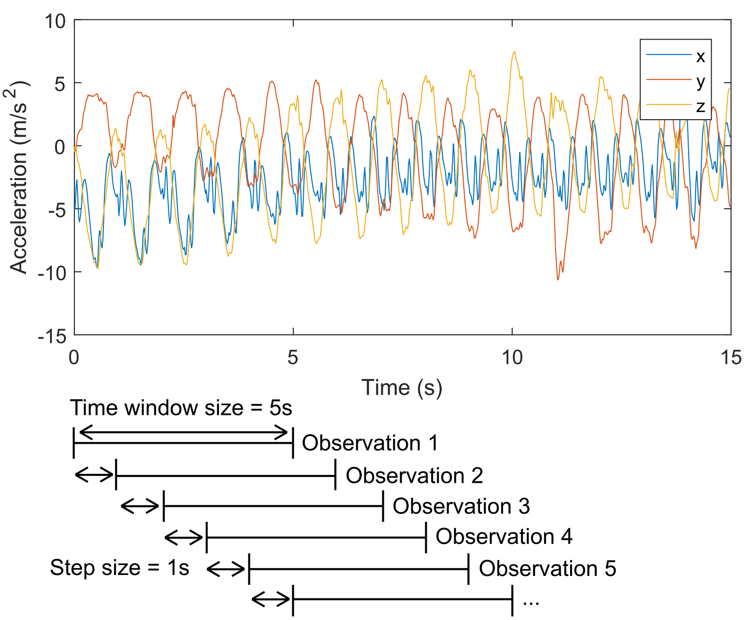

Fig. 2. An example of the segmentation of a data recording.

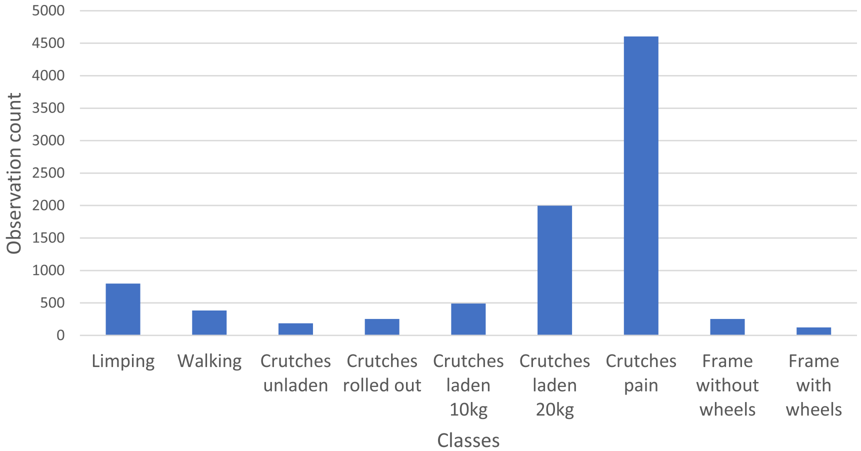

Fig. 3. Observation count of the available dataset of all gait pattern classes.

or no consent from the patient. Figure 3 presents the observation count of the available dataset. To cope with the problem of the imbalanced dataset and to optimize the performance of the classification algorithms, the random minority over-sampling with replacement method was used [18].

We have used Matlab for feature extraction and Python and the Scikit-learn module [19] for machine learning. To evaluate the performance of our system, we split the available dataset into a training set (80%) and a test set (20%) in a stratified fashion. The minority classes of the training set were randomly over-sampled with replacement. The 10-fold cross-validation scheme was used on the training set to train the model, the performance of which was evaluated on the test set. Figure 4

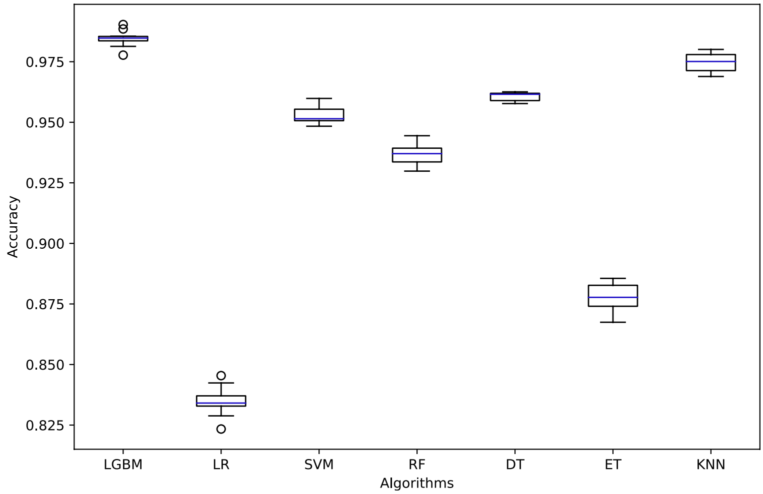

Fig. 4. Box plot of multiple classifiers trained for gait recognition.

Table 3. Confusion matrix of the LGBM classifier.

	Predicted class								
True class	L	W	CU	CRU	CL 10 kg	CL 20 kg	CP	FN	FW
Limping (L)	155	0	0	0	0	0	5	0	0
Walking (W)	3	66	0	0	0	0	8	0	0
Crutches unladen (CU)	0	0	36	0	0	1	0	0	0
Crutches rolled out (CRU)	0	0	0	43	1	2	5	0	0
Crutches laden 10 kg (CL 10 kg)	0	0	0	0	81	3	14	0	0
Crutches laden 20 kg (CL 20 kg)	0	0	0	0	0	377	23	0	0
Crutches pain (CP)	0	2	0	0	1	19	899	0	0
Frame without wheels (FN)	0	0	0	0	0	0	6	44	0
Frame with wheels (FW)	0	0	0	0	0	0	0	0	24

Table 4. Performance metrics of the LGBM classifier.

True class	Precision	Recall	F1-score
Limping (L)	0.981	0.969	0.975
Walking (W)	0.971	0.857	0.91
Crutches unladen (CU)	1	0.973	0.986
Crutches rolled out (CRU)	1	0.843	0.915
Crutches laden 10 kg (CL 10 kg)	0.976	0.827	0.895
Crutches laden 20 kg (CL 20 kg)	0.938	0.943	0.94
Crutches pain (CP)	0.937	0.976	0.956
Frame without wheels (FN)	1	0.88	0.936
Frame with wheels (FW)	1	1	1

presents the box plot for all trained classifiers. Different classifiers naturally perform differently. This is due to the nature of the problem, the characteristics of the dataset and the capacity of each classifier in terms of the variety of functions it can fit. Table 3 presents the confusion matrix for the LGBM classifier, the best performing classifier and Table 4 presents the model's performance metrics.

We have achieved an accuracy of 94.9% with the LGBM classifier on the previously unseen test set. From the confusion matrix, it is worth noting that the misclassified observations belonging to one of the crutches classes were most of the times predicted to belong to another crutches class. Although misclassified *per se*, these kinds of observations may still provide physiotherapists useful information regarding the gait patterns of the patients.

3.3. *Fully connected neural network*

The same dataset that was constructed earlier was fed into a fully connected neural network. To evaluate the performance of our system, we split again the available dataset into a training set (80%) and a test set (20%) and the 10-fold cross-validation scheme was used on the training set to fine-tune the model. The network consists of two hidden layers, the first with 72 neurons and the second with 36 ones, both applying the Rectified Linear Unit (ReLU) activation function. We have used the

Adam optimization algorithm [20] with its default values and since we have a problem of multiclass classification, we used the categorical cross-entropy as the loss function. Last but not least, the batch size was set to ten. We have used Python and the Keras API for the neural networks.

We achieved a performance of 90.9% on the test set. We notice that using the same dataset, the fully connected neural network that we trained performs worse than the LGBM classifier of the previous test.

3.4. *Convolutional neural networks*

For the models we have used so far, we manually engineered features from the time series data based on a fixed time window. However, there are deep learning methods such as recurrent neural networks and one-dimensional convolutional neural networks, that provide competent results with minimal or no feature engineering efforts.

We are exploring how one-dimensional convolutional neural networks perform for our problem of gait pattern recognition. The data that we will feed the neural networks with are the raw time-series ones, as these were produced after the universal resampling step with the sampling frequency of 60 Hz. We have used the same time window of five seconds with a step size of one second. So for every time window, we end up with 300 values per axis per sensor. We were using two sensors and each one of those had three axes, so we end up with six features in total. So each row of data contains 1800 elements. This is 50 times more than the features we manually engineered for the previous tests, so it is very likely that there are some redundant data.

The network that we have built consists of two one-dimensional convolutional neural network layers, both with a standard configuration of 64 feature maps and a kernel size of three. We have added a dropout layer [21] with a value of 0.2 for regularization and to prevent overfitting by slowing down the learning process. Then we have added a pooling layer that reduces the learned features to half of their size in order to avoid overfitting and accelerate the training procedure. After the convolutional network, the learned features are flattened and passed through a fully connected layer of size 100 before the output layer. Last but not least, we are using again the Adam optimizer to optimize the network and the categorical cross-entropy loss function for our multiclass classification problem.

Due to the stochastic nature of neural networks, we repeat the evaluation of the model 20 times and then summarize the performance of the model across those runs. We are training several one-dimensional convolutional neural networks and we discuss how important parameter tuning is when creating such models.

3.5. *Standardization*

One possible transformation that we can apply on the available dataset is to standardize the input before training the model. By standardizing a variable, its distribution is shifted so that it has zero mean and a standard deviation of one.

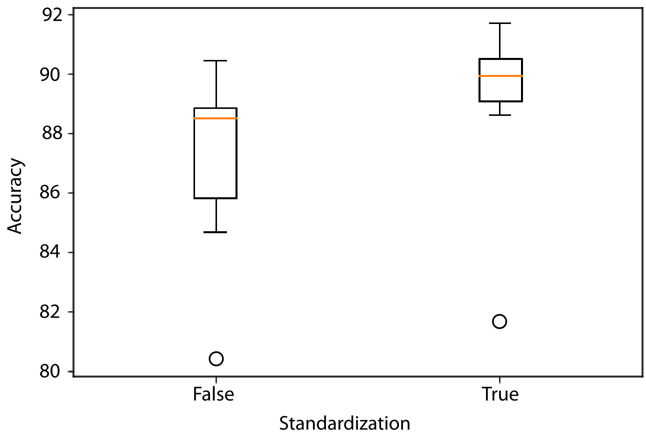

Fig. 5. One-dimensional convolutional neural networks with and without standardization.

We evaluated our model both with and without standardization and the results are presented in Fig. 5.

We can notice that by standardizing our dataset we can easily lift our predictive performance. Without standardization the accuracy was 87.3%, while with standardization that accuracy increased to 89.3%. For this reason, for the next tests we are standardizing the input before training the neural networks. The performance of the convolutional network at this point did not surpass the fully connected one we tested before, since the former did not have the learning capacity to do so as the latter had with the already engineered features. With the next tests we are tuning the hyperparameters of the convolutional network in order to achieve a higher accuracy.

3.6. *Number of filters*

In this test we are exploring how modifying an important hyperparameter of a convolutional neural network such as the number of filters, has an impact on the overall predictive performance of the model. Specifically, we tested the following values for the number of filters: 8, 16, 32, 64, 128 and 256. Figure 6 presents the predictive accuracies for the different number of filter maps.

The bigger the number of filters we use, the better the accuracy gets. We have managed to achieve an accuracy of up to 92% using 128 filters. However, the more filters we use, the more computationally demanding fitting the model gets.

3.7. *Size of kernel*

Another important hyperparameter of the one-dimensional convolutional neural network is the size of the kernel. This basically controls the number of time steps that are taken into account from the input data on each read. For our test, we used the following values for the kernel size: 2, 3, 5, 7 and 11. Figure 7 presents the predictive accuracies for the different values of the kernel size that were tested.

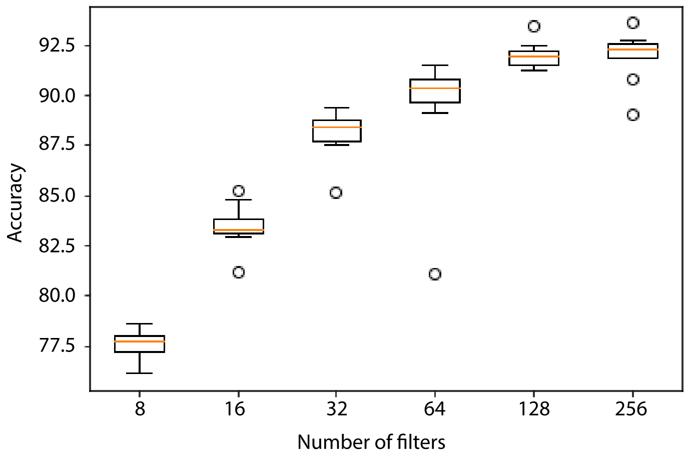

Fig. 6. One-dimensional convolutional neural networks with different number of filter maps.

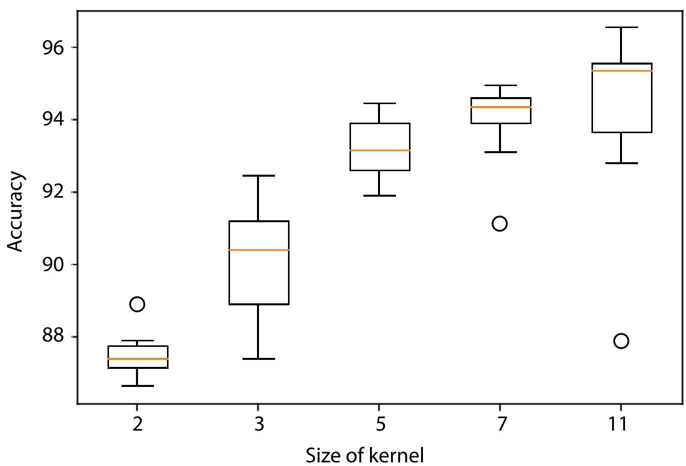

Fig. 7. One-dimensional convolutional neural networks with different kernel sizes.

We achieve the best accuracy for a kernel size of 11 (94.3%). However, the kernel size of 7 provides a better balance between low variance and good performance (94%) and might be a better choice for our case.

At this point, we can notice that even without any feature engineering, we can achieve very good results with a one-dimensional convolutional neural network by optimizing its hyperparameters. Tradeoffs however exist. It is computationally more demanding to train a neural network compared to a gradient boosting model. However, in the long run when new labeled data might become available, having an already trained neural network to retrain might be easier than having another machine learning model that will need retraining from scratch.

4. Conclusion

This paper presented a machine learning-based, gait recognition system that assists physiotherapists with the postoperative rehabilitation phase of patients who have undergone a lower body operation. The architecture of the system comprising a smartwatch, a smartphone, and a web server was presented. The performance of the system was validated with labeled data recorded by physiotherapists of the Hirslanden Clinique La Colline, an orthopedic clinic in Geneva, Switzerland. Gait patterns of patients were recorded soon after they have undergone various types of lower body operations. After engineering time and frequency domain parameters, several machine learning models and a fully connected neural network were tested. The predicted performance of the system reached an accuracy of 94.9% with the best performing classifier among nine different gait classes. One-dimensional convolutional networks were also trained with the raw time-series data of the sensors. After the hyperparameter tuning, a predictive performance of 94.3% was achieved. The innovation of the proposed system lies in the fact that it enables physiotherapists to monitor the evolution of the gait pattern of a patient under rehabilitation, throughout the day and not only during the defined and time-limited physiotherapy sessions.

Acknowledgments

This work was supported by the Swiss National Science Foundation (Contract 100019_165572) in the frame of the From Lab to Life: Cognitive Aging Revisited project and by the Commission for Technology and Innovation CTI, of the Swiss federal government (CTI Grant 18566.2 PFES-ES) in the frame of the Recover@home project.

References

[1] A. M. Moseley, R. D. Herbert, C. Sherrington and C. G. Maher, Evidence for physiotherapy practice: A survey of the physiotherapy evidence database (pedro), *Aust. J. Physiother.* **48**(1) (2002) 43–49.

[2] C. G. Maher, C. Sherrington, R. D. Herbert, A. M. Moseley and M. Elkins, Reliability of the pedro scale for rating quality of randomized controlled trials, *Phys. Ther.* **83**(8) (2003) 713–721.

[3] W. Pirker and R. Katzenschlager, Gait disorders in adults and the elderly, *Wien. Klin. Wochenschr.* **129**(3–4) (2017) 81–95.

[4] C. T. Valadao, F. Loterio, V. Cardoso, T. Bastos, A. Frizera-Neto and R. Carelli, Robotics as a tool for physiotherapy and rehabilitation sessions, *IFAC-PapersOnLine* **48**(19) (2015) 148–153.

[5] M. B. D. Santos, C. B. D. Oliveira, A. D. Santos, C. G. Pires, V. Dylewski and R. M. Arida, A comparative study of conventional physiotherapy versus robot-assisted gait training associated to physiotherapy in individuals with ataxia after stroke, *Behav. Neurol.* **2018** (2018) 1–6.

[6] R. Bodor, B. Jackson and N. Papanikolopoulos, Vision-based human tracking and activity recognition, in *Proc. 11th Mediterranean Conf. Control and Automation*, 2003, pp. 18–20.

[7] L. Bao and S. S. Intille, Activity recognition from user-annotated acceleration data, *Int. Conf. Pervasive Computing*, 2004, pp. 1–17.

[8] A. I. Kyritsis, M. Deriaz and D. Konstantas, Considerations for the design of an activity recognition system using inertial sensors, in *IEEE 20th Int. Conf. e-Health Networking, Applications and Services*, 2018, pp. 1–8.

[9] Y. Bai, G. J. Welk, Y. H. Nam, J. A. Lee, J.-M. Lee, Y. Kim, N. F. Meier and P. M. Dixon, Comparison of consumer and research monitors under semistructured settings, *Med. Sci. Sports Exerc.* **48**(1) (2016) 151–158.

[10] G. Appelboom *et al.*, Smart wearable body sensors for patient self-assessment and monitoring, *Arch. Public Health* **72**(1) (2014) 28.

[11] M. Yang, H. Zheng, H. Wang, S. McClean and D. Newell, Igait: An interactive accelerometer based gait analysis system, *Comput. Methods Programs Biomed.* **108**(2) (2012) 715–723.

[12] H. K. Chan, H. Zheng, H. Wang, R. Gawley, M. Yang and R. Sterritt, Feasibility study on iphone accelerometer for gait detection, in *5th Int. Conf. Pervasive Computing Technologies for Healthcare*, 2011, pp. 184–187.

[13] A. H. Johnston and G. M. Weiss, Smartwatch-based biometric gait recognition, in *IEEE 7th Int. Conf. Biometrics Theory, Applications and Systems*, 2015, pp. 1–6.

[14] A. Nicholson, M. Lowe, J. Parker, S. Lewis, P. Alderson and A. Smith, Systematic review and meta-analysis of enhanced recovery programmes in surgical patients, *Br. J. Surg.* **101**(3) (2014) 172–188.

[15] D. J. Cook, J. E. Thompson, S. K. Prinsen, J. A. Dearani and C. Deschamps, Functional recovery in the elderly after major surgery: Assessment of mobility recovery using wireless technology, *Ann. Thorac. Surg.* **96**(3) (2013) 1057–1061.

[16] C. V. Bouten, K. T. Koekkoek, M. Verduin, R. Kodde and J. D. Janssen, A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity, *IEEE Trans. Biomed. Eng.* **44**(3) (1997) 136–147.

[17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[18] G. Lemaître, F. Nogueira and C. K. Aridas, Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, *J. Mach. Learn. Res.* **18**(17) (2017) 1–5.

[19] F. Pedregosa *et al.*, Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.* **12** (2011) 2825–2830.

[20] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *CoRR* arXiv:1412.6980.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* **15**(1) (2014) 1929–1958.