Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 239 (2024) 177–184

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2023

Real Time Detection of Freezing of Gait of Parkinson Patients based on Machine Learning Running on a Body Worn Device

Ali Haddadi Esfahani [a,*], Oliver Maye[a] , Max Frohberg[a] , Maria Speh[c] , Micheal Jöbges[c] , Peter Langendörfer[a,b]

[a] IHP–Leibniz-Institut für innovative Mikroelektronik, Im Technologiepark 25, 15236 Frankfurt/Oder, Germany
[b] *BTU Cottbus-Senftenberg, 03046 Cottbus, Germany*
[c] *Kliniken Schmieder Allensbach, Zum Tafelholz 8, 78476 Allensbach, Germany*

Abstract

In this work, we present a system that detects Freezing of Gait Detection (FOG) that uses of a single wearable inertial sensor to automatically evaluate a Parkinson's patient's gait instability and detect FOG in real-time. A wearable vibrator is our cueing system which is triggered by the FOG detection whenever a FOG episode takes place. The vibration impulses help the patient to prevent FOG by switching to voluntarily movement execution. Sensor data were collected from nine patients with Parkinson's disease performing Unified Parkinson's Disease Rating Scale (UPDRS) test under the supervision of a clinical expert. Along with data recording, a video was taken from patient's parkour. The data were labeled through the recorded video of the patient's tests and FOG and non-FOG data were assigned. A machine learning model using a deep Long-Short-Term-Memory (LSTM) employ the accelerometer data from the sensor and the inference leads to a FOG or non-FOG classification. The FOG detection model is pruned, quantized and used for real-time inference on Google Coral board worn on the patient' body. The model deployed on a Google coral board sends a trigger to the cuing device right after the FOG detection and the patient get alert for the happening FOG. The individualized model for one-second windows applied in this work performed an average of 91.5% of sensitivity and 86.5% specificity for models running on PC and 91.7% of sensitivity and 86.7% of specificity for the models tested on Google coral with the latency of 50 millisecond on real-time testing.

* Corresponding author. Tel.: +49-3355625-263.
E-mail address: haddadi@ihp-microelectronics.com

Peer-review under responsibility of the scientific committee of the CENTERIS / ProjMAN / HCist 2023
10.1016/j.procs.2024.06.160

Keywords: Freezing of gait; Long short term memory; Machine learning; Edge device; Recurrent neural network; Time-series classification

1. Introduction

Worldwide, Parkinson's disease (PD) affects over 10 million individuals. When a PD patient reaches the intermediate stage of the disease, the risk of falling increases dramatically, from 38% to 68% [1][2][3]. Parkinson's disease is a chronic and progressive neurodegenerative disorder altering several motor and non-motor symptoms typically in elderly people with an age over 60[4][5]. Issues with gait and balance are some of the common motor symptoms of Parkinson's disease that gradually appear over time [6]. For those who have Parkinson's disease, one of the most incapacitating symptoms is the Freezing of Gait (FOG). Gait impairment and disruptions limit everyday activities and reduce quality of daily life along with the increase in the risk of falling [8].

Thanks to recent advancement in embedded electronics and sensors besides their adaptation in the wearable device market, patients with Parkinson's disease can take advantage of real-time and automatic analysis and evaluation of their disease especially in FOG detection. Above all, low power devices are becoming more and more capable running neural networks. This enables researchers to implement complex models on wearable devices that analyses sensor data to detect FOG in real-time, the maximum time before cuing is triggered are 250 msec. Such sensors may be placed where they will collect the most useful information for FOG detection. Convincing results are achieved with sensor worn at the ankle.

Machine learning algorithms can analyze a wide variety of data such as spatial or temporal produced by a wearable sensor. The Machine Learning (ML) models which have already been trained specifically for FOG detection can produce a decision based on input data produced by Inertial Measurement Units (IMU). A proper dataset containing data labeled as FOG and non-FOG plus a correct modeling can produce accurate FOG detection in due time.

This work used supervised machine learning algorithms to automatically find FOG events and train individual models for each patient. The models then figure out how to classify raw data from the IMU sensor as FOG or not FOG. A windowing strategy takes a predefined number of samples that represent a window. Each window has a subset of data that overlaps with its neighbors. In our dataset, there are nine patients who walked through a specific parkour tests (see Figure 1) while being watched by a physiotherapist. Three channels of acceleration data and video-assisted labels were used for training and testing.

While the trained models are ready for use on a PC, they need to be adapted for use in an embedded device. Quantizing and pruning a network, significantly reduce the amount of memory required, so it fits in an embedded device. TensorFlow lite was used for the model conversion.

The main contribution of this paper is that it covers a comprehensive procedure from data gathering to model deployment in a wearable device to detect FOG. The wearable device comes with means to read sensor data and hardware for model inference as well as with a real-time on-demand vibration cuing system. Unprocessed data that represent real-life situations are used to test each patient's individual model.

The mean sensitivity of the PC models are 91.4% while the sensitivity for Google Coral's model is 91.7%. The specificities gained 85.5% on PC models and 85.7% on Google coral models. The models did not show any drop in sensitivity when the models were converted from PC to Google Coral. Similarly, in specificity two models showed a slight growth of non-FOG detection rate while only one model decreased by 1% after pruning and conversion to TensorFlow lite.

This paper is organized as follows. Section 2 discusses related work, the data gathering and the parkour protocol. Section 3 describes LSTM in FOG detection. Section 4 presents the evaluation results of our approach. Section 5 concludes the paper and highlights our future directions.

2. Related works

Literature in the field of FOG detection does not really rely on the same dataset. This limits the possibility to compare the model performance of other works with this one and to produce a benchmark along the research executed so far. In addition, the number of papers which implement their model on an embedded device and use their model in real-time is substantially lower than the one running models on PCs.

LSTM based models are widely employed in FOG detection and human activity recognition due the nature of temporal input and direct sensor reading analysis [14][15][16][17]. Sigcha et al. suggested a CNN and Long Short-Term Memory (CNN-LSTM) deep neural network model to analyze data from a single waist-mounted accelerometer [18]. There were a total of 21 people engaged in the study, and the data was gathered from patients in their own homes to maximize the number of FOG episodes. They used Leave-One-Subject-Out (LOSO) for the model's validation. The authors discovered that the best result could be obtained by stacking three prior spectral windows on the current window as the input of the CNN-LSTM model using a 3.2 seconds-window, yielding a mean sensitivity, specificity, and geometric mean of, all equally, 87.1%.

Bikias et al. have presented a new CNN model to look into the potential of employing a wrist-based IMU sensor to identify FOG events [19]. Data from the CuPiD IMU dataset, which included information on 18 patients, was used for the analysis of FOG episodes. An accelerometer and a gyroscope sampled data at 128 hertz, and a three-second window was used. According to the applied 10-fold cross-validation, a basic network with two CNN layers attained a mean sensitivity of 86%, and specificity of 90%.

One of the major aspects of the above-mentioned studies is that their model was only applied on powerful computation devices such as PCs. There has not been a lot of attention on analyzing the performance of FOG detection systems in real-time or leveraging the potential of hardware that is capable of AI inference and data acquisition.

Kim et al. installed the AdaBoost algorithm on a smartphone and gathered information on the patient's mobility using the smartphone's built-in acceleration sensor [20]. The patient is given auditory cues to serve as an early warning system in the event that FOG develops. The attained inference latency was close to one second, and the model performance was 86% in sensitivity and 91.70% in specificity, respectively.

Smartphones were also utilized by Punin et al. for data collection and processing [21]. Using three mounted sensor nodes and a delay of 750 ms by 15-second windows, their model classifies FOG based on statistical approaches. The sensitivity of the model is 86.7%, while the specificity is 60.6%.

FPGA designs were also adopted to implement ML models for FOG detection [22][23]. Mikos et al. implemented the neural network on an FPGA. They employed their own feature selection methodology to pick the most effective features in their model in order to cope with the limited physical connections available in FPGAs. Their research could achieve profound results in sensitivity and specificity of 95.6% and 90.2%, respectively, deploying a window length of 4.5 seconds resulting in 20ms of inference. Their device can run for around 9 hours with the battery of 800mAh [23]. Langer et al. ran the Temporal Convolutional Neural (TCN) network on Xilinx FPGA and used Daphnet dataset to train and test the data [22]. Their individualized TCN model could classify the window less than 1 millisecond. The very short detection time provides the key factor for cueing devices to trigger and helps the Parkinson's patient to avoid falling. The average sensitivity and specificity achieved for the models running on the FPGA are 78% and 90%, respectively.

3. Data gathering and methodology

3.1. Dataset and experimental design

Our dataset includes nine patients with PD showing considerable variation in motor skills. All of the patients elicited FOG episodes during their tests. Two patients are female while the remaining seven patients are male subjects. Data were collected from wearable FOG detection device with the patients' consent. The data were stored anonymously using a unique numbering format. The parkour protocol was designed and applied to the patients by clinical experts in the Kliniken Schmieder Allensbach. The designed parkour protocol mimics some activities from Bächlin et al. such as rotation, obstacle passing and activity of daily life [24].

The parkour, see Figure 1, is composed of several activity sessions representing various aspects of daily walking. Obstacle passing through the boxes, rotation, walking on rough surface, intentional stop and start orders during parkour are among the major tasks. Each patient repeated the parkour several times each day and also on different days in order to achieve variation of FOG behavior over time. In addition to primary walking the parkour, patients were asked to answer questions to distract them form their major task i.e. walking.

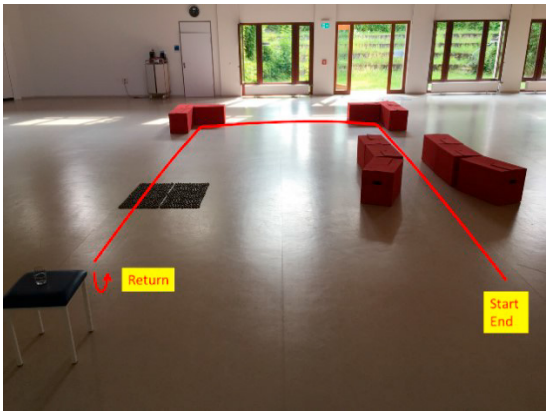

Figure 1: the parkour setup covers rotation of 90° and 180° to the left and right. The assistant orders the patient to stop and start while the patients were answering questions.

The data were recorded while patients walked the parkour and while the clinical expert was observing and recording the patient's parkour on video. The labels were assigned by the clinical expert during a video review.

This research work is based on data from only one acceleration sensor worn on the patient's foot. The FOG detector device collect three axis of acceleration data using Bosch Sensortec BMA456 sensor. The FOG detector sensor was worn at the outer part of ankle of patient's right foot. The data acquisition had 100 Hz of frequency collecting the data from acceleration sensor. A custom-design add-on board to the Coral Mini board was designed and developed by us for this purpose, see Figure 2. The recording firmware was also developed and tested by us.

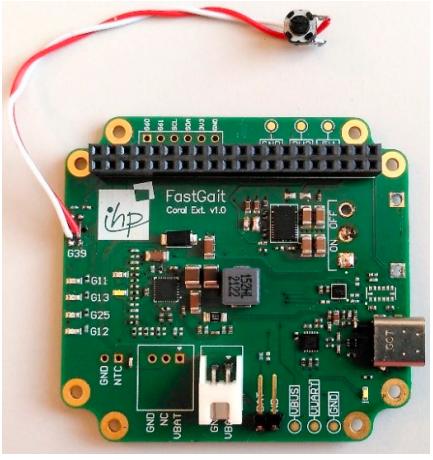

Figure 2: Custom-design add-on data acquisition board, designed and developed by IHP. It connects to the Google coral board.

Once the data is cleaned and prepared for supervised learning, they are ready to be used for training and testing. All the training, and testing processes were done in TensorFlow 2.6. The window length is 1 second containing 100 samples for each channel, during training and inference. A rolling window was applied to the data. The rolling window starts from the beginning of each region, FOG and non-FOG, and consecutive windows have a 50% overlap of data with the adjacent window. In the training set, the last window of a region which is not completely filled with data of

the same label is dropped. In other words, windows can only contain data with one label, and they cannot be partially filled with FOG and non-FOG data. Windows get the label according to their inner data.

3.2. Model Implementation

The raw acceleration data were cleaned and prepared for labeling in MATLAB 2021a. The cleaning process started with removing all three channels of a timestamp if any of the three channels failed to record. Thanks to the robust firmware and customized add-on data acquisition board developed at IHP-Microelectronics, the data recording showed very few data drops, with a rate below 0,001% of the whole data. The add-on labeling app in MATAB was used to assign the region of interest with the help of video recordings of patients' tests and labels provided by clinical expert. Label 1 was assigned to the region of FOG, while label 0 was used for non-FOG regions. The unwanted parts of data which were out of the scope of the parkour protocol such as the recording data before and after the parkour test were removed. Finally, the labeled regions were masked i.e. each timestamp gets a label 0 or 1 representing non-FOG or FOG, respectively.

Once the data are cleaned and prepared for supervised learning, they are ready to be used for training and testing. All the training, and testing processes were done in TensorFlow 2.6. The window length is 1 second containing 100 samples for each channel, during training and inference. A rolling window was applied to the data. The rolling window starts from the beginning of each region, FOG and non-FOG, and consecutive windows have a 50% overlap of data with the adjacent window. In the training set, the last window of a region which is not completely filled with data of the same label is dropped. In other words, windows can only contain data with one label, and they cannot be partially filled with FOG and non-FOG data. Windows get the label of their inner data.

The label of a certain window depends on the majority of the labels of the timestamps represented in the window. For the test set, the FOG or non-FOG label is applied to the windows that contain exactly 100% of FOG or non-FOG data. Windows are only examined for a FOG or non-FOG label by the algorithm if they contain at least 70% of the associated data, otherwise, they are ignored.

Our model is made up of three stacked layers of LSTM blocks. The input data is fed into our LSTM model with 110 hidden units in the first layer, followed by two more layers with 90 and 70 hidden units, respectively. The total number of learnable parameters in the model is 167600. The LSTM layers are followed by a fully connected layer, and then the Softmax layer distributes output probabilities to one of two classes. The last step of the inference is to determine if the window is classified either FOG or non-FOG. The LSTM model is of the sequence-to-one kind, therefore the input data is three-channel raw data from the acceleration sensor positioned on the ankle of the patient, and the output is a label describing the class of the input data.

The Glorot initializer and the orthogonal matrix initializer are used, respectively, for initializing the input weights and the recurrent weights. The LSTM block's cell candidate is given a hyperbolic tangent function. Input, forget, and output gates' nonlinear transformations are all calculated using the sigmoid activation function. Following each mini-batch of training data, the Adam optimizer adjusts the model weights. The initial learning rate was set to 0.007, and the learning algorithm revises it every three epochs. For the Adam solver, the decay rate factor in the gradient moving average is 0.83. The definition of the squared gradient moving averages is 0.99. These values yielded the highest model performance throughout the trials.

The training procedure was modified using ridge regression. The factor of weight decay is set at 0.0011. Overfitting is prevented using early stopping mechanism by terminating training early when the validation set is 10% poorer than the training set. If the early stopping criterion is met for three iterations in a row, training is terminated. 30 epochs is the maximum allowed. The windows were randomly selected for training and test set. 80% were used for training, 10% for validation and the last 10% for testing. An equal distribution of labels among each set was preserved. In order to level off the disparity between the FOG and non-FOG data in the training set, the FOG windows were increased three times, so that FOG data was increased from 15.3 percent in the original dataset to 45.9 percent after the augmentation.

4. Results and discussion

4.1. Model size and latency

The data type of the LSTM model parameters was assigned float32 which stores both the raw data and model parameters in the PC version. The model was converted to the Coral board using the optimizer and pruning procedure of TensorFlow lite and the datatypes of learned parameters were converted into float16. Memory requirements for inference on PC and Google coral board, as well as the number of computed values inside the LSTM block, are listed in Table 1. For PCs, the trained LSTM model is over 4 megabytes in size. When compressed and pruned, the trained LSTM model shrank from 4133 Kilobytes (KB) on PC to 181 KB on the Google coral board, a 96% decrease in size. The pruned model which is applicable on Google coral board takes remarkable little space, i.e. 181 KB, of the 8 MB SRAM of google coral board. The whole inference procedure, beginning with the feeding of a window containing 100 samples and ending with the classification of the window by assigning a label to it, the inference takes around 50 milliseconds. The timing requirements for FOG detection on a wearable device are fully met by the achieved inference time.

Table 1: The number of parameters of LSTM model, the model size is in kilobyte (KB) on PC and Google Coral board, and Inference latency on Google Coral.

Learned parameters	Internal calculated parameters	Total parameters	Model size on PC (KB)	Model size on Google coral board (KB)	Inference latency on Google Coral board (ms)
167600	3193	170793	4133	181	50

4.2. Model performance

This research relies on unique models for each patient. Each of the models of a patient, both the trained model on a personal computer and the converted models on a Google coral board, were tested using the same test set from the patient in order to maintain consistency in the analysis. The robustness of the model is measured using assessment measures such as sensitivity and specificity, which shows the detection rate of FOG to that of non-FOG.

Results from real-time testing showed that the patient-dependent models for one-second windows developed in this work had a mean sensitivity of 91.5% and a specificity of 86.5% when running on a PC, and average sensitivity of 91.7% and specificity of 86.7% when tested on Google coral with a latency of 50ms, only. Almost all of the patient specific models attained sensitivity higher than 85%, and even more impressive, three models achieved a FOG detection rate of more than 98%. Out of the nine patients, only one patient had a performance that was below average, patient 4, whose model only had 73% sensitivity. This patient barely showed FOG in his overall test, only 3% of overall recording, and the FOGs lasted less than seconds. Because there were so few FOG data, we were left with very little information, which resulted in a less accurate model for patient 4, see Figure 3.

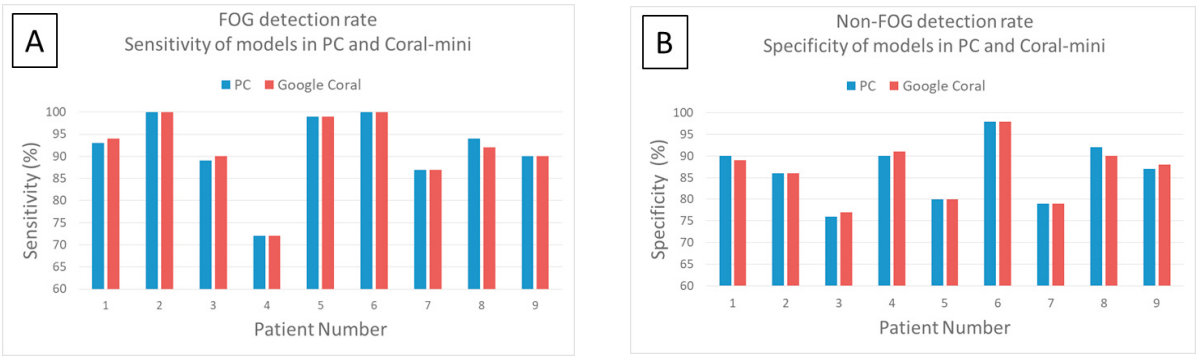

Figure 3: LSTM model performance for FOG (A) and Non-FOG detection (B) rates.

The compressed model that is compatible with Google coral board improves the overall performance of the majority of the models in terms of both sensitivity and specificity, see Figure 4. Two models' sensitivity increased by 1% while one model had a drop of 2% after the conversion into the TensorFlow Lite model, see Figure 4[A]. The conversion of the model had the similar impact on the specificity of the data as well. After the conversion using TensorFlow Lite, the non-FOG detection rate was the same for four models, while three models showed improvements of 1%, two models showed 1% and 3% drop of specificity, see Figure 4[B].

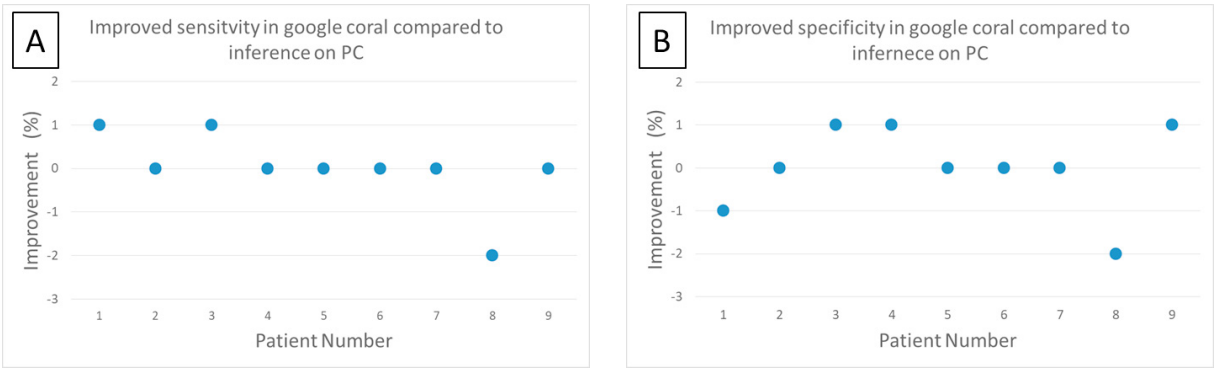

Figure 4: model performance improvement after conversion to TFLite model.

The fact that even though the model size was reduced by approximately 96%, the majority of the models showed similar performance in their FOG and non-FOG detection rate, is counter-intuitive. Our current explanation is that this effect is due to the quantization and converting dense matrices into sparse ones. This needs further investigations.

When compared to previous research, such as that of Sigcha et al., which utilized windows of 3.2 seconds and that of Bikias et al., which used windows of 3 seconds, we chose a window length that was significantly shorter i.e. just one second [19]. Mikos et al. and Punin et al. are two of the papers that have lengthy windows, with lengths of 4.5 and 15 seconds, respectively. Both of these studies have implemented their model on an embedded device [23][21]. The window length that was used in the research carried out by Kim et al., which is 1 second, is most comparable to this work in terms of both its length and its latency [20]. On the other hand, the algorithm was executed on a smartphone, i.e. on a powerful device. Our FOG detection device is an embedded device i.e. it is resource constraint which means running ML algorithms is challenging. In addition its design does not require user interaction in contrast to the smartphone.

Our work is comparable with the research results discussed above, but showed better sensitivity and better FOG detection rate compared to Punin et al., Kim et al., and Bikias et al. and Sigcha et al. [21][20][19][18]. Our window length of 1 second, is also much smaller than the 4.5 seconds used in Mikos et al. [23], leading to faster classification result from the beginning of a window to the decision, which is essential for fall prevention.

5. Conclusion and outlook

This paper demonstrated the feasibility of FOG detection using a body worn device in real-time. A parkour protocol was defined and the acceleration data was acquired under the clinical supervision. Our dataset was labeled and prepared for training and testing using a deep LSTM network. The models were quantized, pruned, and prepared to be deployed on Google coral board for real-time inference application.

To the best of our knowledge our paper is the first research work running the ML model for FOG detection on Google coral hardware. Our model used, three channels of acceleration sensor data and 1-second windows to detect FOG and non-FOG events. Our patient-dependent models achieved an average, 91.5% and 86.5% of sensitivity and specificity, respectively, on PC and obtained similar results i.e. 91.7% and 86.7% of sensitivity and specificity on the Google coral device, respectively. The model classified input data in 50ms.

In our future work we will take into account that Parkinson's disease is neuro degenerative process which increases severity of the disease over time. A model assessment needs to analyze the robustness of model performance at

different points in time such as after 3 or 6 month using the same parkour. We will apply post-training using data from the patient and update the models of each patient with newer data.

References

[1] Roller WC, Glatt S, Vetere-Overfield B, Hassanein R. Falls and Parkinson's Disease. Clin Neuropharmacol 1989;12:98–105. https://doi.org/10.1097/00002826-198904000-00003.

[2] Macht M, Kaussner Y, Möller JC, Stiasny-Kolster K, Eggert KM, Krüger H-P, et al. Predictors of freezing in Parkinson's disease: A survey of 6,620 patients. Mov Disord 2007;22:953–6. https://doi.org/10.1002/mds.21458.

[3] Chen P-H, Wang R-L, Liou D-J, Shaw J-S. Gait Disorders in Parkinson's Disease: Assessment and Management. Int J Gerontol 2013;7:189–93. https://doi.org/10.1016/j.ijge.2013.03.005.

[4] Sveinbjornsdottir S. The clinical symptoms of Parkinson's disease. J Neurochem 2016;139:318–24. https://doi.org/10.1111/jnc.13691.

[5] Ahn D, Chung H, Lee H, Kang K, Ko P, Kim NS, et al. Smart Gait-Aid Glasses for Parkinson's Disease Patients. IEEE Trans Biomed Eng 2017;64:2394–402. https://doi.org/10.1109/TBME.2017.2655344.

[6] Triarhou LC. Dopamine and Parkinson's Disease. Madame Curie Biosci. Database, Austin (TX): Landes Bioscience; 2013.

[7] Fleyeh H, Westin J. Extracting body landmarks from videos for Parkinson gait analysis. Proc - IEEE Symp Comput Med Syst 2019;2019-June:379–84. https://doi.org/10.1109/CBMS.2019.00082.

[8] Ren K, Chen Z, Ling Y, Zhao J. Recognition of freezing of gait in Parkinson's disease based on combined wearable sensors. BMC Neurol 2022;22:1–13. https://doi.org/10.1186/s12883-022-02732-z.

[9] Mileti I, Germanotta M, Di Sipio E, Imbimbo I, Pacilli A, Erra C, et al. Measuring gait quality in Parkinson's disease through real-time gait phase recognition. Sensors (Switzerland) 2018;18:1–16. https://doi.org/10.3390/s18030919.

[10] Giladi N, McDermott MP, Fahn S, Przedborski S, Jankovic J, Stern M, et al. Freezing of gait in PD: Prospective assessment in the DATATOP cohort. Neurology 2001;56:1712–21. https://doi.org/10.1212/WNL.56.12.1712.

[11] Schaafsma JD, Balash Y, Gurevich T, Bartels AL, Hausdorff JM, Giladi N. Characterization of freezing of gait subtypes and the response of each to levodopa in Parkinson's disease. Eur J Neurol 2003;10:391–8. https://doi.org/10.1046/j.1468-1331.2003.00611.x.

[12] Arias P, Cudeiro J. Effect of Rhythmic Auditory Stimulation on Gait in Parkinsonian Patients with and without Freezing of Gait. PLoS One 2010;5:e9675. https://doi.org/10.1371/journal.pone.0009675.

[13] Young WR, Shreve L, Quinn EJ, Craig C, Bronte-Stewart H. Auditory cueing in Parkinson's patients with freezing of gait. What matters most: Action-relevance or cue-continuity? Neuropsychologia 2016;87:54–62. https://doi.org/10.1016/j.neuropsychologia.2016.04.034.

[14] Shalin G, Pardoel S, Lemaire ED, Nantel J, Kofman J. Prediction and detection of freezing of gait in Parkinson's disease from plantar pressure data using long short-term memory neural-networks. J Neuroeng Rehabil 2021;18:1–15. https://doi.org/10.1186/s12984-021-00958-5.

[15] Esfahani AH, Dyka Z, Ortmann S, Langendorfer P. Impact of Data Preparation in Freezing of Gait Detection Using Feature-Less Recurrent Neural Network. IEEE Access 2021;9:138120–31. https://doi.org/10.1109/ACCESS.2021.3117543.

[16] Masiala S, Huijbers W, Atzmueller M. Feature-Set-Engineering for Detecting Freezing of Gait in Parkinson's Disease using Deep Recurrent Neural Networks 2019.

[17] Ordóñez FJ, Roggen D. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors (Switzerland) 2016;16. https://doi.org/10.3390/s16010115.

[18] Sigcha L, Costa N, Pavón I, Costa S, Arezes P, López JM, et al. Deep learning approaches for detecting freezing of gait in parkinson's disease patients through on-body acceleration sensors. Sensors (Switzerland) 2020;20. https://doi.org/10.3390/s20071895.

[19] Bikias T, Iakovakis D, Hadjidimitriou S, Charisis V, Hadjileontiadis LJ. DeepFoG: An IMU-Based Detection of Freezing of Gait Episodes in Parkinson's Disease Patients via Deep Learning. Front Robot AI 2021;8:1–8. https://doi.org/10.3389/frobt.2021.537384.

[20] Kim H, Lee HJ, Lee W, Kwon S, Kim SK, Jeon HS, et al. Unconstrained detection of freezing of Gait in Parkinson's disease patients using smartphone. Proc Annu Int Conf IEEE Eng Med Biol Soc EMBS 2015;2015-Novem:3751–4. https://doi.org/10.1109/EMBC.2015.7319209.

[21] Punin C, Barzallo B, Huerta M, Bermeo A, Bravo M, Llumiguano C. Wireless devices to restart walking during an episode of FOG on patients with Parkinson's disease. 2017 IEEE 2nd Ecuador Tech Chapters Meet ETCM 2017 2018;2017-Janua:1–6. https://doi.org/10.1109/ETCM.2017.8247520.

[22] Langer P, Haddadi Esfahani A, Dyka Z, Langendörfer P. FPGA-Based Realtime Detection of Freezing of Gait of Parkinson Patients, 2022, p. 101–11. https://doi.org/10.1007/978-3-030-95593-9_9.

[23] Mikos V, Heng CH, Tay A, Yen SC, Chia NSY, Koh KML, et al. A Wearable, Patient-Adaptive Freezing of Gait Detection System for Biofeedback Cueing in Parkinson's Disease. IEEE Trans Biomed Circuits Syst 2019;13:503–15. https://doi.org/10.1109/TBCAS.2019.2914253.

[24] Bächlin M, Roggen D, Tröster G, Plotnik M, Maidan I, Hausdorff JM, et al. Wearable assistant for Parkinsons disease patients with the freezing of gait symptom. IEEE Trans Inf Technol Biomed 2010;14:436–46. https://doi.org/10.1109/TITB.2009.2036165.