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

Ali Haddadi Esfahani [1,*]
Wireless systems and applications
haddadi@ihp-microelectronics.com

Oliver Maye[1]
Wireless systems and applications
maye@ihp-microelectronics.com

Max Frohberg[1]
Wireless systems and applications
frohberg@ihp-microelectronics.com

Maria Speh[3]
m.speh@kliniken-schmieder.de

Michael Jöbges[3]
m.joebges@kliniken-schmieder.de

Peter Langendörfer[1,2]
langendoerfer@ihp-microelectronics.com

1 Introduction

For those who have Parkinson's disease, one of the most incapacitating symptoms is 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 [1]. Thanks to recent advancement in embedded electronics and sensors as well as their adaptation in the wearable device market, low power devices are becoming more and more capable running neural networks. This enables researchers to implement complex models on wearable devices that capture and analyze sensor data to detect FOG in real-time.
In this study, a single wearable inertial sensor is used to automatically analyze the gait instability of a Parkinson's disease patient and identify FOG in real-time. The dataset used was recorded while a patient was walking the parkour. The parkour protocol was designed and applied by clinical experts in the Kliniken Schmieder Allensbach. The parkour consists of a number of actions depicting different aspects of everyday walking. In the parkour, some of the most important activities are obstacle traversal via the boxes, rotation, walking on a rough surface, and stopping and starting on purpose. The data were collected in the parkour tests while the clinical expert observed and recorded videos of the patient walking through the parkour. During the video evaluation, the clinical expert assigned labels corresponding to FOG episodes. The ethical approval procedure of tests and data gathering of the PD's patients was handled by Klinikum Schmieder, and granted before the experiments.
In this work we used supervised machine learning algorithm using a deep Long-Short-Term-Memory (LSTM) to train an individual model for each patient and to classify raw data from the IMU sensor as FOG or not FOG. Unprocessed data that represent real-life situations are used to test the patient's individual model.
TensorFlow Lite was used to convert the trained PC model into one, compatible with a Google coral device [2].
In order to avoid FOG after its detection cueing needs to be triggered. Our cueing system is a wearable haptics setup and its vibrator is triggered whenever a FOG episode occurs, see Figure 1.
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 and trigger cueing.

* Corresponding author: Ali Haddadi Esfahani (e-mail: haddadi@ihp-microelectronics.com).
[1]IHP–Leibniz-Institut für innovative Mikroelektronik, Im Technologiepark 25, 15236 Frankfurt/Oder, Germany
[2]BTU Cottbus-Senftenberg, 03046 Cottbus, Germany
[3]Kliniken Schmieder Allensbach, Zum Tafelholz 8, 78476 Allensbach, Germany

This research has been partially funded by the Federal Ministry of Education and Research of Germany under grant number 03ZZ5301E.
CHASE '23, June 21–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0102-3/23/06.
https://doi.org/10.1145/3580252.3589423

2 Dataset and Model Implementation

The wearable device developed in this work uses data from only one acceleration sensor worn on the patient's foot. The FOG detector sensor was worn at the outer part of ankle of the patient's right foot. The data acquisition had a frequency of 100 Hz when collecting the data. The FOG detector device collects three axis of acceleration data using the Bosch Sensortec BMA456 sensor. A custom-design add-on board was designed and developed by us for this purpose. The recording firmware was also developed and tested by us.

3 Results and discussion

Table I shows the memory requirements for inference on PC and Google coral board, as well as the number of calculated

Table I: 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

values inside the LSTM block. The trained LSTM model for PCs exceeds 4 MB in size. When compressed and pruned, the trained LSTM model decreased in size by 96%, from 4133 Kilobytes (KB) on PC
to 182 KB on the Google coral board. Beginning with the feeding of a window comprising 100 samples and concluding with the classification of the window by assigning a label, the whole inference operation takes around 50 milliseconds. The obtained inference time meets the timing criteria for FOG detection on a wearable device, which is 250 milliseconds.

In order to ensure consistency in the study, both the trained model on a PC and the converted models on a Google coral board were evaluated using the same test set from the patient. 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, respectively.

Results from real-time testing showed that the patient-dependent models for one-second windows developed in this work had a sensitivity of 93% and a specificity of 94% when running on a PC, and a sensitivity of 90% and a specificity of 89% when tested on Google coral. Even though the size of the converted model was reduced by approximately 96%, the Google coral's model had only 3% reduction in sensitivity. Similar behavior is seen in the specificity with a difference of 5% from the PC model to the converted model. Our current explanation is that this effect is due to the quantization and converting dense matrices into sparse ones. This needs further investigation.

4 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 clinical supervision. Our dataset was labeled and prepared for training and testing using a deep LSTM network. The models were converted to be deployed on Google coral 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 patient-dependent model used three channels of acceleration sensor data and 1-second windows to detect FOG and non-FOG events. The model classified input data in 50ms. The most important aspect is that our system was able to release the patient from the FoG in 100% of the detected FoG situations. In our future work, we will consider more data from other patients and evaluate their model performance on the board- and PC-basis models.

Figure 1: Google coral mounted on shank (upper). The cueing device, lower image, receives the trigger command whenever the FOG detected the ML model on Google Coral.

REFERENCES

[1] Kang Ren, Zhonglue Chen, Yun Ling, and Jin Zhao. 2022. Recognition of freezing of gait in Parkinson's disease based on combined wearable sensors. *BMC Neurol.* 22, 1 (2022), 1–13. DOI:https://doi.org/10.1186/s12883-022-02732-z

[2] TensorFlow. TensorFlow Lite. Retrieved April 17, 2022 from https://www.tensorflow.org/lite