2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Deep Learning Based End-to-End Locomotion Mode Detection Method for Lower Limb Wearable Robot Control

Zeyu Lu[1], Ashwin Narayan[2] and Haoyong Yu[3]

Abstract— **To function effectively in real-world environments, powered wearable robots such as exoskeletons and robotic prostheses must recognize the user's motion intent by detecting the user's locomotion modes such as walking, stair ascent and descent or ramp ascent and descent. Traditionally, intent detection is achieved using rule based methods such as state machines or fuzzy logic using data from wearable sensors. Due to the difficulty of manual rule design, these methods are limited to detect certain simple locomotion modes. Machine learning (ML) based methods can perform classification on a large number of classes without manual rule design and recent research has explored several ML methods for locomotion mode classification. However, current ML based methods for locomotion mode detection use classical methods that require use of feature engineering to achieve acceptable accuracies. Additionally, current ML strategies only classify when certain motion events are detected. This strategy, while computationally efficient could result in misclassifications affecting large sections of motion recognition. To overcome these limitations, this paper proposes an end-to-end deep learning based method for locomotion mode detection that eliminates the need for feature engineering and classifies at a fixed sample rate. This paper introduces a new metric called confidence index and proposes a strategy for tuning confidence index thresholds to achieve a stable intent recognition and overall accuracy of greater than 95% on a publicly available benchmark dataset.**

Keywords— **Lower limb wearable robots, Intent recognition, Machine learning, Locomotion mode**

I. INTRODUCTION

The technology of wearable robots have been well applied to individuals with disability [1]–[4]. Powered wearable robots such as exoskeletons and robotic prosthetic devices need to detect the user's motion intent to effectively assist users in real-world environments. Based on the terrain, human locomotion can be classified into modes that share similar patterns of motion across human. Examples of locomotion modes include walking, stair ascent, stair descent, running, ramp ascent and ramp descent (Figure 1). By identifying these locomotion modes in real-time using data from sensors such as inertial measurement units (IMUs) [5] and electromyography (EMG) [1], [6], wearable robots can adapt their control strategies and patterns of assistance to the specific locomotion mode. In other words, they use the locomotion mode as an estimation of the user's motion intent. Most wearable robots track locomotion modes using rule based methods such as state machines [7], [8], fuzzy logic [9] or decision trees [10]. However, rule based methods for locomotion mode detection require deep study of experimental data for the design of the rules. Rule design becomes increasingly complex and labour intensive for larger numbers of locomotion modes. Consequently, most rule based locomotion mode detection methods are limited to certain simple locomotion modes such as walking.

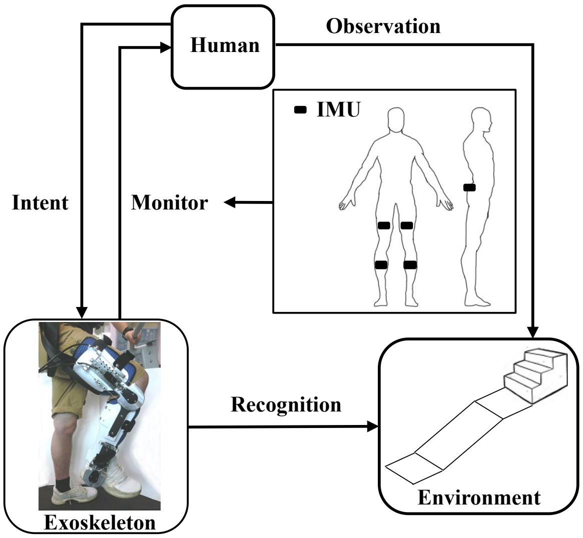


Fig. 1. The human-environment-robot (HER) loop consists of the human locomotion monitor, human intention towards robot, human observation and robotic recognition towards environment. In the human locomotion monitor branch, machine learning models are utilized to recognize IMU signals, which functions as the highest level in controller system of exoskeleton.

Due to their ability to perform large multi-class classification tasks, machine learning based methods have recently gained traction for locomotion mode detection. Recent works have explored the use of several classical machine learning methods such as Support Vector Machines (SVM) [1], [5], [11], Linear Discriminant analysis (LDA) [2], Quadratic Discriminant Analysis (QDA) [6] and Artificial Neural Networks (ANN) [5]. While these methods can be used to detect locomotion modes without the manual design of rules, they make use of feature engineering. Feature engineering is the process of designing transformations that project the input data into a space where the classes of interest are more easily separable by these methods. Classical machine

[1]Zeyu Lu is with the Department of Biomedical Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077 zeyu.lu@u.nus.edu

[2]Ashwin Narayan is with the Department of Biomedical Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077 ashwin_narayan@u.nus.edu

[3]Haoyong Yu is with the Department of Biomedical Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077 bieyhy@nus.edu.sg

This work was supported by Agency for Science, Technology and Research, Singapore, under the National Robotics Program, with A*star SERC Grant No.: 192 25 00045, and by the FRC Tier 1 Grant with WBS No. R397-000-302-114, National University of Singapore.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

learning methods are less effective for data with complex decision boundaries and feature engineering is often needed to achieve acceptable performance.

Many machine learning based methods also perform classification of locomotion modes only when specific motion events are detected [5], [11]–[15] such as heel strike or toe off [16]. These event based methods have lower latency at these gait events. They are also computationally efficient as gait events usually happen at low rates and uneven intervals. However, wrong classifications could potentially lead to much larger segments of activity being wrongly classified compared to continuous classification at a regular rate.

To overcome the limitations of existing locomotion mode classification algorithms, this paper proposes a deep learning based end-to-end classification method for classifying locomotion modes from wearable sensor data. Unlike classical methods, the proposed method is trained to classify locomotion modes end-to-end using raw sensor data without computing any specific feature. Unlike event based methods, the proposed method runs continuously at a fixed sample rate and generates a classification at each iteration or time step.

One of the disadvantages of a continuous intent classifier is that the predicted class can switch between different locomotion modes when the classification probability of the intents is uncertain. A new metric, the confidence index, is proposed that measures how certain the classifier is in the output. By producing output classes only when the confidence index is high, we achieve stable locomotion mode classification with minimal false mode switches.

The proposed method is trained on a benchmark dataset for human lower limb locomotion mode detection and evaluated on key metrics against state of the art existing methods. The results presented indicate that the proposed method is capable of performing end-to-end classification of locomotion modes at fixed sample rates.

II. Methods

A. Dataset

We use a public benchmark dataset [17]. We subscribe the IMU partitioning in the dataset. The dataset contains inertial sensor data from 10 able-bodied (AB) subjects. In total, five 6-DOF IMUs are fixed on their waist, bilateral thigh and shank respectively. The subjects are asked to freely ambulate over level walking (LW), stair ascent/descent (SA/SD) with a four-step staircase and ramp ascent/descent (RA/RD) with 10 degree slope.

B. IMU Data Preprocessing

In the dataset, each data sample contains the signal of 5 IMU sensors. Each IMU records 6 channels of data (3 axis acceleration and 3 axis angular velocity). Every 25 samples ($50ms$), a window of 150 samples ($300ms$) is captured. The choice of window length refers to [5]. The resulting window of data $x \in \mathbf{R}^{150\times30}$ is used as the input to the classifiers. Thus, the sampling rate is 20 Hz. To visualize the inertial signal difference between the five locomotion modes, we compute the means and standard deviations across windows for each mode (Figure 2).

For using with the classical machine learning methods that we compare against (e.g., LDA, QDA, SVM), we also compute features from these windows. The features computed include the mean, standard deviation (SD), maximum, minimum, initial, and final values for each of the 30 channels in the data. This follows the features computed in [5], [7].

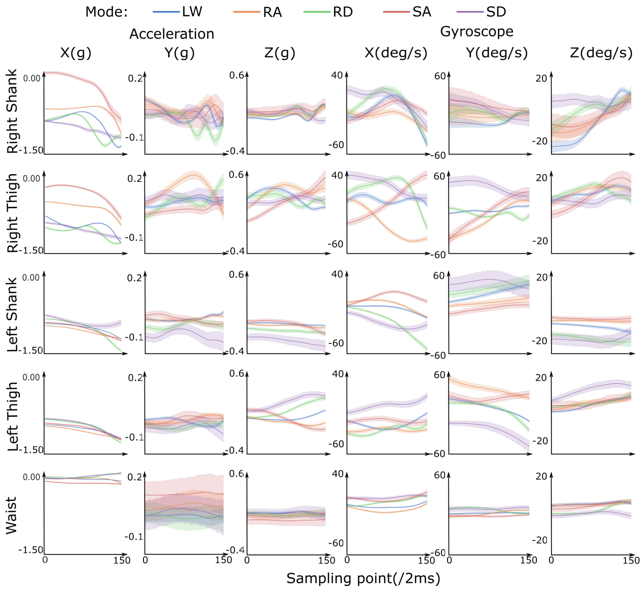


Fig. 2. The lower limb inertial signal from the dataset is divided into windows of 300ms length (150 sampling points). The data is aligned by the order of sampling points (x-axis). The interval between neighbor windows that are adjacent is $50ms$. All the windows are statistically analyzed with computing mean values (solid curve) and variance (shade area). The dataset are collected from signals of 5 IMUs. Each IMU has three channels of acceleration signal and three channels of gyroscope signal. The curve colors vary with different locomotion modes, where LW denotes level walking, RA/RD denote ramp ascent/descent, SA/SD denote stair ascent/descent.

C. Convolutional Layers

The proposed deep learning model's primary building block consists of convolutional layers. Convolutional layers convolute a set of kernels over the input matrix to generate one feature map for each kernel. A non-linear activation function is often applied after the convolution [18], such that

$$x_n = \delta(w_n * x_{n-1} + b_n) \qquad (1)$$

x_n represents n^{th} convolution layer's output and δ is the non-linear activation function. w_n and b_n are the weights and biases of n^{th} convolutional layer respectively.

The Rectified Linear Unit (ReLU) is an activation function that is commonly used in deep neural networks due to their computational efficiency and easier propagation of gradients through the network. The convolutional layers of this model adopt this activation function.

$$\delta(x) = max(x, 0) \qquad (2)$$

D. Fully Connected Layer

The fully connected layer or dense layer applies a matrix multiplication with a set of weights and adds a bias vector before applying a non-linear activation function. As the model classifies input to several mutually exclusive classes, the last layer applies a softmax activation function. The inputs of the softmax activation layer a_i are the outputs of fully connected layer. The outputs of the softmax activation layer $\rho(a_i)$ are classification possibilities that sum up to 1. N is the number of the locomotion modes to be classified. $N = 5$ for our case.

$$\rho(a_i) = \frac{e^{a_i}}{\sum_{n=1}^{N} e^{a_n}}, \text{for } i = 1, ..., N \tag{3}$$

E. Network Architecture

The proposed convolutional neural network (CNN) model (see Figure 3) applies a 1-D convolution (across the time axis) with appropriate padding to preserve the dimensions of the input data. 1D convolutional layers applies a convolution using a 1D kernel over each channel. After the first layer, the network branches in two with each branch consisting of four convolutional layers. Since the features in a sliding window contribute to classification, we would like to preserve the original kinematic information. Therefore, there is no pooling layer since the information loss would be too significant. The proposed CNN model contains a convolutional layer and fully connected layer with softmax layer to extract deep features from sliding windows which is the input and output of the probability for each class.

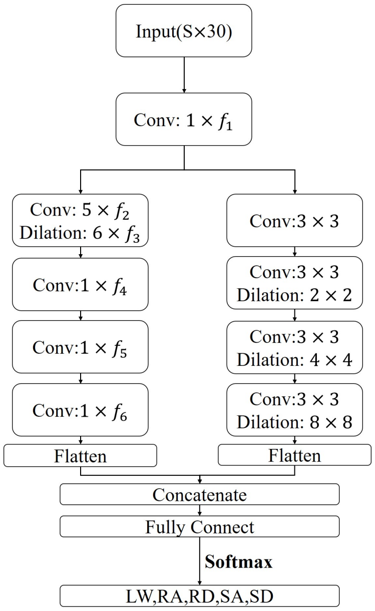


Fig. 3. The neural network architecture for continuous locomotion mode detection. $f_1 - f_6$ are parameters that can be tuned with S, S is the length of the input window.

- IMU Axis Features Separate Branch
  Each IMU has six signal axis consisting of three signal channels of gyroscope and three signal channels of acceleration. Each signal axis points to locomotion modes-wise features (Figure 2). In order to avoid mixture among the features of each axis (e.g., accelerations of x axis), a $5 \times f$ (5 IMUs setup) convolutional kernel with dilation rate of $6 \times f$ was used to extract features from each axis of IMU. After separating signal channels in each IMU, the following layers are 1-D convolution kernel for extracting time-series features and reducing the size of the feature map. Flatten layer would flatten all the outputs from previous layer into a row for concatenating with the flatten outputs of another branch.
- Local and Global Dilation Branch
  This branch is proposed to extract local and global features with 2-D convolutional kernel and ascending size of dilation rate. A bigger dilation rate can help networks to obtain more global feature, because the convolution range increases with dilation rate. The channels could form a connection by dilated kernels. All the outputs at the end of this branch would be flattened and concatenated with outputs of another branch.

F. Mode Specific Classifiers (MSC) Configuration

Rather than using a single classifier to perform classification at each sample, we adopt a mode specific classification approach similar to the one presented in [5]. We train five copies of the deep neural network presented in the sections above, where each copy is corresponding to each of the five locomotion modes considered in this paper. Each copy is trained to classify locomotion data sample that is in the corresponding locomotion mode. The classifier corresponding to each mode differs in the output layer. This constrains the possible mode transitions to a small set and improves accuracy as proved by our results. Table I shows the output classes available to each mode specific classifier. This reduces the size of each individual classifier and improves its computational efficiency.

TABLE I
THE OUTPUT OF EACH MODE SPECIFIC CLASSIFIER FOR LOCOMOTION-MODE DETECTION

Mode Specific Classifier	Output
LW	**LW** RA RD SA SD
RA	LW **RA**
RD	LW **RD**
SA	LW **SA**
SD	LW **SD**

G. Confidence Index

We propose a metric called the confidence index (CI) that measures the average level of prediction confidence of the classifier. If any of the class outputs are highly certain (have a high probability), the confidence index is close to 1.

$$CI = \frac{2 \times \sum_{i=1}^{N} |\rho(a_i) - 50\%|}{N}, \sum_{i=1}^{N} \rho(a_i) = 100\% \qquad (4)$$

where the N denotes the number of locomotion modes to be classified. $N = 5$ for our case. $\rho(a_i)$ is the output probability of each class i, and full mark of CI is 1. According to properties of classifier, every possible class would be given a probability $\rho(a_i)$.

H. Metrics

- Accuracy
  One of the metrics for performance evaluation is accuracy. It is defined as the ratio of the number of correct classifications over the total number of data points.

$$Accuracy = \frac{N_{correct}}{N_{total}} \times 100\% \qquad (5)$$

- Response Time
  The response time is defined as the time taken by the classifier to classify each sample. Response time is an important metric to measure the classifier's performance.

I. Prediction/Inference

To predict locomotion from the neural network's outputs, we use a modified inference procedure. Neural networks with a softmax layer predict the output class that corresponds to the softmax output node with the largest probability.

We compute the confidence index of outputs P_i of softmax layer and only make a prediction of the output class once the confidence index is above a certain threshold. This ensures that the neural network only predicts a change of the locomotion mode if the change is confidently certain. Figure 4 demonstrates the classifier output when different thresholds are specified for the confidence index.

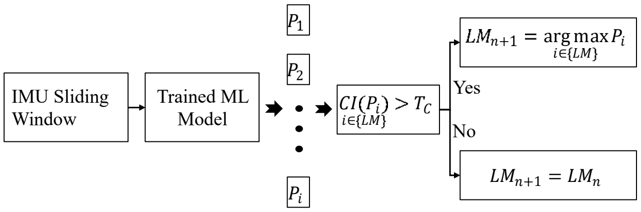

IMU Sliding Window
Trained ML Model
P_1
P_2
P_i
$CI(P_i) > T_C$, $i \in \{LM\}$
Yes
No
$LM_{n+1} = \arg\max_{i \in \{LM\}} P_i$
$LM_{n+1} = LM_n$

Fig. 4. Decision Making Process of Locomotion Mode (LM) Recognition, T_C denotes confidence threshold.

III. Experiment

A. Offline Locomotion Mode Detection by MSC-CNNs

The CNNs model is built using Keras library, and trained on a NVIDIA RTX 2080 Ti. The loss function is the categorical cross-entropy. The training adopts the Adam Optimizer [19] due to its computationally-efficiency. The learning rate was set 10^{-5} or 10^{-4} during the training process

The input of MSC-CNNs (Section II-E & II-F) is the sliding window containing 150×30 features values ($S = 150$) and parameters in our experiments are that $f_{1:6} = \{15, 5, 6, 25, 15, 5\}$, which results in $21k$ trainable parameters in the networks. Then, input is convoluted by a deep neural networks with 1-D, 2-D and dilated 2-D kernels to extract the in-depth features. We separately trained five CNNs models corresponding to five locomotion modes for building MSC-CNNs. A callback for early stopping is activated after five epochs without validation $f1_{score}$ reducing, and the best weight would be stored. The validation error is defined as:

$$Error = 1 - f1_{score} \qquad (6)$$

where the $f1_{score}$ is the mean of precision and recall for each classification result.

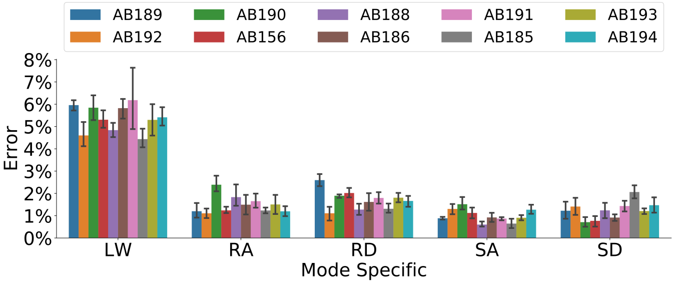


Fig. 5. Validation error across 10 subjects (subject independent) by 5-fold cross-validation.

We implemented the proposed MSC-CNNs classifiers to recognize the locomotion data on the benchmark dataset mentioned in Section II-A and compared performance with previous approaches. To evaluate the proposed confidence threshold, we offline tested the end-to-end recognition filtered by several confidence threshold values. The offline detection experiments were tested on the raw locomotion data of last about 10% locomotion data of subject AB189, while the MSC-CNNs models were trained on the rest 90% data.

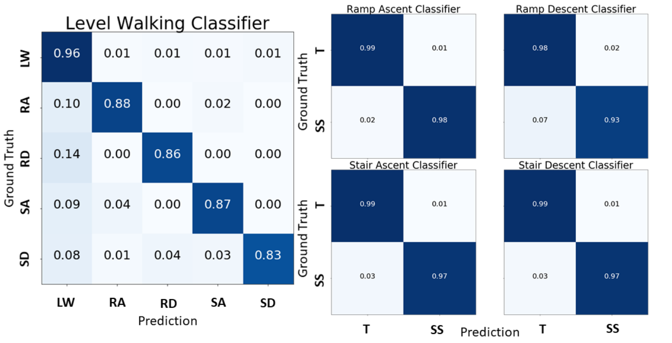


Fig. 6. The accuracy confusion matrix of each MSC classifier. SS denotes steady status, which means keeping the current locomotion mode. T denotes transition, which means switching the locomotion mode to LW.

B. Comparative Study

In this section, we compared the performance of all the machine learning classifiers (i.e., LDA, QDA, SVM, MSC-CNNs and non MSC-CNNs) in terms of overall accuracy cross 10 subjects. All of IMU signals were rolled into sliding windows with a step size of $50ms$ $(20Hz)$ and a window length of 150 samples $(300ms)$, as mentioned in Section II-B. The total number of input frames for training and evaluation is showed in Table II. The evaluation process included: 1) shuffling all frames, 2) applying a stratified 5-fold cross-validation.

TABLE II

THE NUMBER OF SLIDING WINDOWS FROM IMU SIGNALS THAT WERE USED TO TRAIN AND EVALUATE CLASSIFIERS.

Classifier	LW	RA	RD	SA	SD	Total
LW	172080	45907	49777	20033	20281	308078
RA	172080	45907				217987
RD	172080		49777			221857
SA	172080			20033		192113
SD	172080				20281	192361

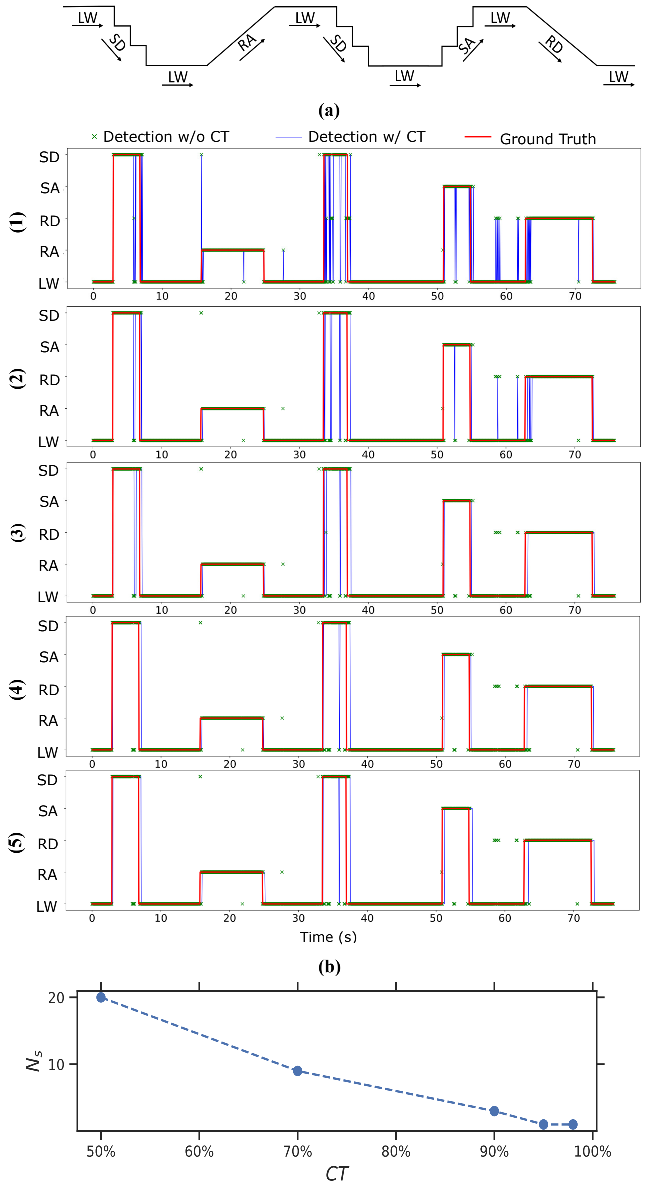
In the comparative study, MSC-CNNs and non MSC-CNNs were implemented as mentioned in Section III-A. For LDA, QDA and SVM's setup using sklearn python package, hyper-parameters of LDA were set as: solver is singular value decomposition, hyper-parameters of QDA was set as: all default, hyper-parameters of SVM was set as: kernel is Radial Basis Function (RBF), $C = 10$ and one-vs-rest strategy.

IV. RESULTS

To validate the proposed model that achieves continuous intent recognition, we first evaluated the offline performance of the model, which simulates the real 5 terrain walking conditions by classifying the unseen locomotion data. Meanwhile, we compared the effect of various proposed confidence threshold on the recognition stability. Then, we tested classical machine learning to compare the performance.

A. Offline Locomotion Mode Detection by MSC-CNNs

Figure 5 shows the validation error of each classifier over data of 10 subjects. The performance of LW classifier is worse than the other four mode specific classifiers' while LW classifier has more locomotion modes to classify than other four classifiers do (Table I). As shown in Figure 7, the detection curve filtered by the confidence threshold (CT) is smoother than the curve without CT threshold, which indicates that CT is effective at removing the spikes due to wrong detections that happen in steady state (no transition) period. Next, the number of wrong spikes against CT suggests that adjusting proposed CT can make intent recognition more stable. The number of spikes is reduced from 20 to 1 when CT is increased from 50% to 98%. Also, the classification performance on the tested samples is reported by the confusion matrix (Figure 6). The influence of misclassification in steady state to wearable robotics will be discussed in next section.



(a)

(b)

(c)

Fig. 7. **(a)** Terrains of signals that segmented for offline detection test. Arrows show the ambulating direction. Data is from one subject (AB189), LW denotes level walking, RA/RD denote ramp ascent/descent, SA/SD denote stair ascent/descent. **(b)** The classic 5-modes context offline detection test on raw IMU data of last about 2 trails on subject AB189, which was implemented by Mode Specific CNNs at varied confidence threshold(CT). As the results shown, CT is able to cancel most of wrong spikes, but the transition delay would increase with CI: **(1)** $CT = 50\%$,**(2)** $CT = 70\%$, **(3)** $CT = 90\%$, **(4)** $CT = 95\%$, **(5)** $CT = 98\%$. w/ CT denotes with confidence threshold, w/o CT denotes without confidence threshold. **(c)** The number of wrong spikes N_s that occur in steady state are reduced with increasing CT.

B. Comparative Study

The comparative study results (Figure 8) include the accuracy ($p < 0.001$), confidence index ($p < 0.001$) tested in the 5-fold cross-validation over each subject and the average response time ($p < 0.001$) for each frame.

The performance of the proposed MSC-CNNs (Mode Specific Classifiers with CNNs) to recognize the IMU signal

TABLE III
THE PERFORMANCE OF MSC-CNNS

Approach	Sensor Type	Classifier	Monitor Type	Monitor Rate	Subjects	Mean Overall Accuracy [std] %
Hu et al. [5]	IMU	LDA	Event Based	-	10	97.74[0.23]
Liu et al. [11]	EMG	SVM	Event Based	-	4	93.5 - 95.5
Proposed	IMU	**MSC-CNNs**	Continuously	**20Hz**	10	**98.06[1.7]**

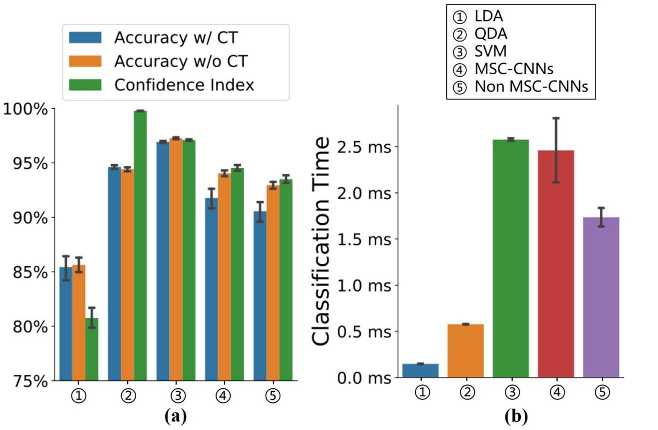


Fig. 8. The comparative study results of linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), support-vector machine (SVM), MSC-CNNs and non MSC-CNNs. **(a)** Overall accuracy and confidence index ($mean \pm SD$). **(b)** Average Response time per frame ($mean \pm SD$). The results are averaged across 10 subjects. w/CT denotes with confidence threshold, w/o CT denotes without confidence threshold.

is evaluated offline through 10 subject on the dataset. **Table III** shows the results of overall accuracy of locomotion recognition. The performance of MSC-CNNs outperforms the Non MSC-CNNs in terms of accuracy and confidence index. Because MSC-CNNs is a novel attempt with continuous locomotion mode detection, we only compared two representative previous works [5], [11]. However, those previous works using state machines for rehabilitation devices make it difficult to implement continuous classification. As a result, proposed work achieved 20Hz monitor rate and outperformed in terms of mean overall accuracy.

V. DISCUSSION

In this study, we demonstrated the potential to detect users intent for lower limb wearable robots. Currently, intent detection strategies either need feature engineering efforts, or are event based potentially causing large segment of misclassification. Few research has been focused on developing a method that can recognize raw signal and perform an event detection. To address those limitations, an end-to-end deep learning based method was evaluated offline on a benchmark dataset. Results showed that our method can perform a high accuracy (above 90%) for classifying raw locomotion signals into multiple human movement such as walking, stair ascent and descent or ramp ascent and descent.

Also, we compared the performance of continuous intent recognition among classical ML methods, which is the part of contribution of this work (Figure 8). QDA, SVM and proposed CNN methods are all above 90% in terms of accuracy and confidence index. However, QDA and SVM need engineering work before classification. When being applied to actual wearable robots, intent recognition controller pursues stability when classifier is accurate enough since any misclassification during steady state may cause serious consequences to user. Therefore, we evaluated the proposed confidence index, and proved its benefit to reducing instability during steady state by removing wrong classification spikes (Figure 7), while the overall accuracy with threshold is decreased slightly. Also, although the latency for entering transition is enlarged when using CT, this is a interesting issue to be optimized for future work. For controller of actual wearable robot, transition latency is still a challenging metric, where people use outreach device, such as camera, that can recognize environment before locomotion [20]. The evaluation results also showed that MSC with multi-classifier outperforms Non MSC with single classifier in terms of accuracy and confidence index.

We believe that our novel approach could contribute to the design of user intent recognition controller of wearable robots system. Although continuous intent recognition probably need much more computational resources than classical event-based state machines, it is reported that the NVIDIA Jetson Xavier can run deep neural networks with large input sizes at rates above 50 Hz in a real-time fashion [21], [22]. Therefore, it is feasible to embed this algorithm in a portable controller and to execute continuous intent recognition in real-time.

VI. CONCLUSION AND FUTURE WORK

This paper evaluated an end-to-end deep neural network, which takes raw IMU data as as input, for locomotion mode detection at a fixed sample rate of $20Hz$, and introduced a new metric called confidence index, and proposed a strategy to tune confidence index thresholds to achieve a more stable intent recognition. This approach achieved a mean accuracy above 90% on a publicly available benchmark dataset. The comparative study results showed that the algorithm achieved performance that is comparable with the state of the art classifiers. Furthermore, it eliminates the needs of feature engineering and avoids large segments with mis-classification.

The proposed algorithm was tested offline on a public dataset. Therefore, future works should include embedding the end-to-end intent detection methods on an actual wearable robotic system and testing the performance of the method in real-time control.

References

[1] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, and K. B. Englehart, "Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion", *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 10, pp. 2867–2875, 2011.

[2] L. J. Hargrove, A. M. Simon, R. Lipschutz, S. B. Finucane, and T. A. Kuiken, "Non-weight-bearing neural control of a powered transfemoral prosthesis", *Journal of NeuroEngineering and Rehabilitation*, vol. 10, no. 1, p. 62, 2013.

[3] G. Aguirre-Ollinger, A. Narayan, and H. Yu, "Phase-synchronized assistive torque control for the correction of kinematic anomalies in the gait cycle", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 11, pp. 2305–2314, 2019.

[4] G. Aguirre-Ollinger, A. Narayan, H.-J. Cheng, and H. Yu, "Exoskeleton control for post-stoke gait training of a paretic limb based on extraction of the contralateral gait phase", in *Biosystems & Biorobotics*, Springer International Publishing, Oct. 2018, pp. 294–298.

[5] B. Hu, E. Rouse, and L. Hargrove, "Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities", *Frontiers in Robotics and AI*, vol. 5, p. 78, 2018.

[6] K. H. Ha, H. A. Varol, and M Goldfarb, "Volitional control of a prosthetic knee using surface electromyography", *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 144–151, Jan. 2011.

[7] H. A. Varol, F. Sup, and M. Goldfarb*, "Multiclass real-time intent recognition of a powered lower limb prosthesis", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 3, pp. 542–551, 2010.

[8] G. Elliott, A. Marecki, and H. Herr, "Design of a clutch–spring knee exoskeleton for running", *Journal of Medical Devices*, vol. 8, no. 3, 2014.

[9] A. Parri, K. Yuan, D. Marconi, T. Yan, S. Crea, M. Munih, R. M. Lova, N. Vitiello, and Q. Wang, "Real-time hybrid locomotion mode recognition for lower limb wearable robots", *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2480–2491, 2017.

[10] M. Liu, D. Wang, and H. H. Huang, "Development of an environment-aware locomotion mode recognition system for powered lower limb prostheses", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 434–443, 2015.

[11] M. Liu, F. Zhang, and H. H. Huang, "An adaptive classification strategy for reliable locomotion mode recognition", *Sensors*, vol. 17, no. 9, 2017.

[12] G. Chen, P. Qi, Z. Guo, and H. Yu, "Gait-event-based synchronization method for gait rehabilitation robots via a bioinspired adaptive oscillator", *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 6, pp. 1345–1356, 2017.

[13] M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. d. R. Millán, R. Riener, H. Vallery, and R. Gassert, "Control strategies for active lower extremity prosthetics and orthotics: A review", *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, p. 1, 2015.

[14] E. Zheng, Q. Wang, and H. Qiao, "Locomotion mode recognition with robotic transtibial prosthesis in inter-session and inter-day applications", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 9, pp. 1836–1845, 2019.

[15] J. A. Spanias, A. M. Simon, S. B. Finucane, E. J. Perreault, and L. J. Hargrove, "Online adaptive neural control of a robotic lower limb prosthesis", *Journal of Neural Engineering*, vol. 15, no. 1, p. 016 015, 2018.

[16] J. Taborri, E. Palermo, S. Rossi, and P. Cappa, "Gait partitioning methods: A systematic review", *Sensors*, vol. 16, no. 1, 2016.

[17] B. Hu, E. Rouse, and L. Hargrove, "Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals", *Frontiers in Robotics and AI*, vol. 5, p. 14, 2018.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. eprint: `arXiv:1412.6980`.

[20] K. Zhang, J. Luo, W. Xiao, W. Zhang, H. Liu, J. Zhu, Z. Lu, Y. Rong, C. W. de Silva, and C. Fu, "A subvision system for enhancing the environmental adaptability of the powered transfemoral prosthesis", *IEEE Transactions on Cybernetics*, pp. 1–13, 2020.

[21] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. eprint: `arXiv:1512.03385`.

[22] M. Vandersteegen, K. Vanbeeck, and T. goedeme, *Super accurate low latency object detection on a surveillance uav*, 2019. eprint: `arXiv:1904.02024`.