MoveNet: A Deep Neural Network for Joint Profile Prediction Across Variable Walking Speeds and Slopes

Rishabh Bajpai and Deepak Joshi

***Abstract*—An exoskeleton needs reference joint angle profiles at various speeds and slopes for its application in real-world scenarios. Recording these profiles and their implementation in the control system of an exoskeleton are time-consuming and complex processes. Therefore, there is a need for an artificial system that can predict subject-specific joint angle profiles for various real-world scenarios from a minimum amount of input data. This study aims to propose a predictive neural network (NN) called MoveNet, for joints angle profile prediction across variable walking speeds and slopes. MoveNet consists of three NN modules, namely, encoder, mapper, and decoder. The encoder module is trained to convert the knee joint angle profile into its 6-D latent representation. The mapper module is trained to map the latent representation of angle profile at zero degrees angle of inclination (AOI) to latent representation of angle profile at the target AOI. The decoder module is trained to predict an angle profile from its given latent representation. The proposed model successfully predicted knee joint angle profiles at three walking speeds (0.8, 1, and 1.2 m/s) and nine AOI ranging from −10° to 10°. MoveNet obtained root mean squared error of 3.24 ± 1.19° and mean absolute error of 2.66 ± 1.00° from tenfold cross validation. These results suggest that artificial intelligence can predict subject-specific knee joint angle profiles for variable slopes at a given walking speed from the data of knee joint angle profiles recorded at a flat surface.**

***Index Terms*—Automation, human locomotion, predictive models, robot control.**

I. Introduction

UNDERSTANDING the biomechanics of human locomotion across various slopes and walking speeds has been of great importance for future generation prosthesis and exoskeletons. Such understandings may lead to adaptive behavior of the assistive devices for the activity of daily living (ADL) [1], [2]. One of the key factors in adaptations of those devices, including prosthesis and exoskeleton, is the accurate and continuous prediction of lower limb joint profiles. The knee joint being one of the most weight-bearing joints has been the focus of prosthesis and exoskeleton designers [3]–[6]. In addition, the knee joint provides significant torque during walking, running, and squatting [7], [8]. An accurate knee angle estimation actuates the motors on the knee joint of the prosthesis and exoskeleton to provide positive mechanical work at the joint [9], [10]. Various methods have been evolved for estimating knee joint profiles during overground walking [11]–[14]; however, very few have reported the estimation across inclined surfaces with various slopes [15], [16]. In general, a model is trained for every possible slope of the inclined surfaces for joint trajectory estimation in a real-world scenario. Although useful, this approach requires extensive data from an individual to train the model, while the model walks on different slopes and speeds. Further, such models are not appropriate for extrapolation or interpolation of knee joint profile estimation outside the trained inclined surfaces. Furthermore, such extensive data collection in an amputee population is reported difficult by the amputees, leading to their unwillingness to participate in the experiments [17]. Therefore, a generalized model may potentially reduce the input data required to estimate the knee joint angle profiles across various slopes.

In addition to walking on different inclinations, change in the walking speed is a natural phenomenon during ADLs, and therefore, it should be taken care of while designing future generation prosthesis and exoskeleton. Previous works have reported knee joint profile estimation in able-bodied individuals while walking with self-selected speeds [18]–[20]. Our recent work has estimated the knee joint profiles with three different walking speeds, slow, medium, and fast, in able-bodied and amputee populations [21]. A limited number of reports are available where knee joint angle estimation across different slopes with variable walking speeds is reported [15], [16].

A. Related Work

All the related works can broadly be divided into two categories based on the method used for generating reference trajectories: neural network (NN)-based methods and polynomial-based regression (PR) models. This section presents the work related to both categories sequentially. Cunha *et al.* [22] compared the performance of two NNs, namely, the backpropagation neural network (BNN) and the extreme learning machine (ELM) for generating subject-specific joint angle reference profiles. These NNs were trained to predict hip joint and knee joint reference profiles from the given four subject-specific parameters (SSPs) as

Manuscript received March 8, 2021; accepted April 8, 2021. Date of publication April 16, 2021; date of current version May 10, 2021. The Associate Editor coordinating the review process was Adam G. Polak. *(Corresponding author: Deepak Joshi.)*

The authors are with IIT Delhi, New Delhi 110016, India, and also with the All India Institute of Medical Sciences Delhi, New Delhi 110029, India (e-mail: bmz208129@iitd.ac.in; joshid@iitd.ac.in).

Digital Object Identifier 10.1109/TIM.2021.3073720

input. These parameters included the subject's height, weight, age, and desired walking speed. This study used gait data of 42 subjects and achieved excellent generation accuracy. However, the four SSPs used as input for the models may not represent all subject-specific gait properties. A better approach was proposed by He *et al.* [23]; they considered both predefined SSP and normal gait joint profile as an input of their proposed NN. Their NN-based model employed an ELM module and an autoencoder, which extracted SSP from the given natural gait joint profile. Liang *et al.* [24] proposed a similar approach of using natural gait joint profile as input for their NN. They used a long-short term memory (LSTM) NN for generating hip joint and knee joint gait profiles for a limb using joint angle reference profiles of the other three limbs. Although the generation accuracy of their method was excellent; however, their method cannot be used for a subject having either diplegia or hemiplegia. Liu *et al.* [25] proposed a gait trajectory generation methodology using the deep spatial–temporal model to predict future knee joint trajectories based on the past gait data of the hip joint and the ankle joint. This deep spatial–temporal model consisted of recurrent neural networks (RNN) and LSTM memory units. This model successfully learned the spatial–temporal relationship between the knee joint and other joints. However, this relationship was only tested on a flat surface, which limits its usability during ADLs. Similar to the work of Liu *et al.* [25], Błażkiewicz and Wit [26] proposed an NN that used the gait joint profiles of hip joint and ankle joint to predict the knee joint profiles. In contrast to the preceding work, this study used real-time gait profiles of hip joint and ankle joint. Ferreira *et al.* [27] compared the performance of the three commonly used models for joint profile generation, namely, ELM, ANN, and multi-output support vector regression. They found that ELM performed better than others when SSP was used for generating knee angle profiles at various walking speeds in the range of 0.28–1.53 m/s.

Embry *et al.* [16] proposed the most relevant work to this study, in the class of PR models. They developed a predictive model that represented gait kinematics as a continuous function of gait cycle percentage, speed, and incline. They showed that the performance of their method was significantly better than the linear interpolation technique. However, a large amount of data, including joint profiles at various conditions, is needed for fitting the basis polynomial functions used in their model. This requirement of joint profiles data at various conditions for training the model limits the usability of the model in real-world applications. Fukuchi and Duarte [28] proposed a second-order polynomial relationship between given walking speed and gait cycle points of the knee joint. They tested their model on walking speed in a range of 0.39–2.2 m/s. Ali *et al.* [29] used a quintic polynomial equation and a cubic polynomial equation for generating knee joint trajectory. Though their method predicted joint profiles accurately, their method failed to predict knee profiles for walking at variable speeds and slopes. Moreover, Moissenet *et al.* [30] developed a multiple regression model that takes walking speed, gender, age, and BMI of the subject to reproduce joint profiles. This proposed approach estimated the relative contribution to angular variations of each predictor, independently of the others. However, a better generation performance is expected from a multiple regression model. A similar study from Hong *et al.* [31] used a Gaussian process regression that took walking speed and SSP for generating kinematics at different walking speeds. As the authors claimed, their method obtained better accuracy than the other state-of-art methods at a given walking speed. Smith *et al.* [32] proposed a linear and quadratic PR model that established relationships of walking speed, cadence, and stride length of a subject with its kinematics. Moreover, Ren *et al.* [33] proposed a personalized gait generation method based on 14 SSPs. Popular machine learning regression algorithms namely, random forest, support vector regression, NN, and eXtreme gradient boosting, were exploited for hip joint, ankle joint, and knee joint profile generation. Recently, Bao [34] exploited a CNN-LSTM hybrid model for joint kinematics prediction using surface electromyography.

B. Motivation and Contributions

The previous studies addressed in Section IV tried to predict user-specific joint profiles for the exoskeletons using various methods. Most of these methods used SSP as features for training their models. However, the development of gait in an individual is not purely dependent on its anthropometry, weight, age, gender, BMI, and walking speed of the gait. In fact, gait is a complex sensorimotor behavior heavily affected by cognitive and affective aspects [35]. Therefore, handcrafted SSP would not capture the full characteristic properties of the gait. There are a few more complications of using handcrafted SSP for gait prediction. Since gait is a complex sensorimotor behavior, we need a large number of SSP to capture its full characteristic properties. However, it is not feasible to compute and test the correlation of a large number of SSP with gait cycle angles. Also, the computation involved in defining the complex handcrafted SSP will require a significant amount of effort. A better approach can be the use of an NN that can automatically extract complex features associated with the lower limb joint profiles. Furthermore, these methods also need a lot of data recorded from the user and a lot of time and energy for calibrating these models' parameters. The process of recording the gait data from a person having gait abnormalities becomes several times more complicated than the healthy subjects. This process can be painful and irritating for children having cerebral palsy. Although ADL includes various combinations of walking speeds and slopes, however, most of the previous studies neglected to predict joint profiles for different slopes. Thus, there is a need for a more versatile predictive model that can construct lower limb joint reference profiles for various conditions, including multiple combinations of walking speeds and slopes from the minimum amount of input data.

After considering the limitations of the previous methods in detail, this study presents a novel NN named MoveNet that solves both the problems of the availability of data at different conditions and the requirement of low generation error of the system for real-life applications. This model accepts a joint angle profile of one gait cycle recorded at a flat surface and predicts the joint angle profile for the required angle of inclination (AOI) at the given walking speed. Furthermore, the predicted joint profiles are highly specific to the user and their selected gait speed. There are three modules of the proposed model: encoder module, mapper module, and decoder module. The encoder module is trained for learning a low-dimensional representation of the given joint profile that can represent both global and SSPs. On the other hand, the decoder module is trained for reproducing the high-level representation of the joint profile from its given low-level representation. The third module, the mapper module, is trained for learning a mapping between two low-level representations of joint profiles according to the given AOI of the target joint profile. To achieve the required user-specific joint angle profile, these modules are trained in two phases, and leave-one-out cross validation is performed. In the first phase (Phase I), encoder and decoder modules are trained, and in the second phase (Phase II), the mapper module is trained.

The rest of the work is organized as follows. Data preprocessing, network architecture, training of the network, and performance evaluation metrics are described in Section II. Results of the proposed model are presented in Section III. The significance of the research, future plans, and utilization of MoveNet in powered exoskeleton is discussed in Section IV. Finally, the conclusion of the work is drawn in Section V.

II. Methodology

A. Data Set Description

In this study, a publically available gait data set [16], [36] recorded on ten able-bodied subjects (five females and five males; mean ± SD; age: 23 ± 2.8 years; height: 170 ± 8.2 cm; and weight: 64 ± 7.7 kg) with written informed consent and ethical approval obtained by the Institutional Review Board at The University of Texas at Dallas is used. This selection of subjects represents a population of young, active, and healthy walkers. As per [16], able-bodied is referred to a population of young, active, and healthy walkers. The data set consists of three constant walking speeds of 0.8, 1, and 1.2 m/s and nine AOIs ranging from $-10°$ to $+10°$ at an increment of 2.5°, recorded on a Bertec instrumented split-belt treadmill [16]. Each walking speed has nine conditions for AOIs. Thus, there are $3 \times 9 = 27$ different conditions for each subject. For each condition, the gait data are recorded for 60 sec at a sampling rate of 100 Hz by a ten-camera Vicon motion capture system [16]. A gait cycle is a sequence of events during regular locomotion of a human in which one foot contacts the ground to when that same foot again contacts the ground. It involves propulsion of the center of gravity in the direction of motion. These sequence of events of the gait cycle are referred to as “gait cycle instances,” throughout this article. The kinematic data of one gait cycle are converted into a vector of length 151, where the first 150 points represent interpolated gait cycle instances and the 151st point represents the duration of the gait cycle. There are three reasons for choosing a fixed length of the vector as 151. The first is to deal with variable durations of the gait cycle. The second is to ensure stable training of MoveNet by converted raw kinematic data into gait cycle instances as each neuron in the first layer of the network is associated with a gait cycle instance. The third is that the benchmarking method on this data set also used a vector of the same size. The authors try to provide a fair comparison with the benchmarking method by keeping most of the preprocessing steps the same. The original joint profile can be obtained by rescaling interpolated gait cycle events with the cycle’s duration. Sample knee joint profiles (first 150 points of the vector) for a subject at $-10°$, 0°, and 10° AOIs are shown in Fig. 1.

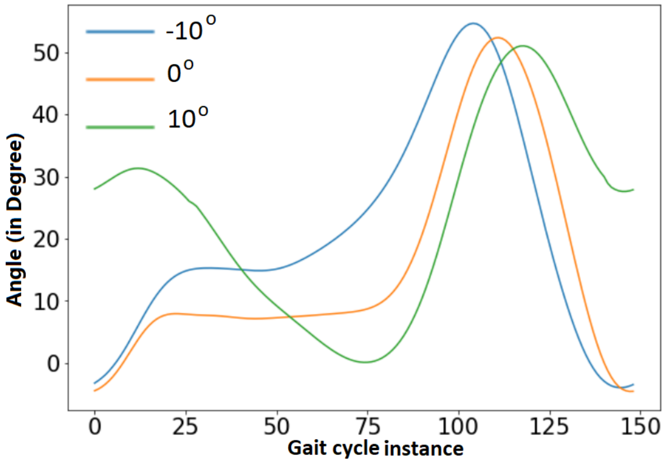

Fig. 1. Sample plot of knee angle in sagittal plane at variable slopes. Here, zero-gait-cycle instance corresponds to 0% of the gait cycle, and 150-gait-cycle instance corresponds to 100% of the gait cycle.

Due to the wide range of motion of knee angle in the sagittal plane compared to other joints’ angle, it is challenging to predict its profile using an NN [37], [38]. This motivated us to develop an NN for knee angle in the sagittal plane. However, the proposed NN without any modification can be used for any other lower limb joint.

For further processing, the MATLAB variables present in the data set are converted into python dictionaries. For training phase I, we define an input sample point as a vector having length 151 as explained above. In this phase, both input and output samples of the NN are the same. For training phase II, we define an input sample point as two vectors having length 151 and length 1. These two vectors having length 151 and length 1 contain information of joint profile at 0° AOI (151 values) and AOI of the target joint profile (a float value), respectively. The output sample point is defined as a vector having a length of 151. It contains information on the joint profile at the given target AOI. Padding of zeros on both sides of the gait vectors is added to make their dimensions equal to 192. This helped in symmetric max-pooling and upsampling operations used in the proposed model. Thus, the dimensions of the input data matrix and output data matrix for phase I are $N \times 192$ and $N \times 192$, respectively. The dimensions of the input data matrices and output data matrix for phase II are $[M \times 192 + M \times 1]$ and $M \times 192$, respectively. Here, N and M having values 27 106 and 26 559 respectively, are the total number of sample points used for training, validating, and testing of the model in their respective phases. For phase I, all gait cycles are used; therefore, the value of N is equal to 27 106, i.e., the number of cycles for all subjects. For phase II, gait cycles at 0 degree AOI condition are used as the input. All gait cycles for each condition at a certain walking speed are averaged. These averaged gait cycles are used as the output for phase II. Thus, all combinations of input and output give the value of M equal to 26 599.

B. Architecture of MoveNet

There are three parts of MoveNet: encoder, mapper, and decoder. Encoder and decoder modules are trained simultaneously in phase I of training, while the mapper module is trained separately in phase II of training by freezing the weights of the other two modules. Fig. 2 illustrates the architecture of MoveNet and its modules.

The encoder has five convolutional layers with the same filter size and different dimensions. Their filter size is 3, and the values of their dimension are 64, 32, 16, 8, and 1. Here, the zero padding is used during the convolution operation for maintaining the same dimensions as the previous layer. LeakyReLU is used as the activation function after each convolution operation. The activation function adds nonlinearity to the system and helps the neural network in learning complex nonlearn features. Max-pooling with a filter size of 2 is applied to the activated convoluted volumes for downsampling. There are six neurons (nodes) in the bottleneck of this autoencoder for representing the input joint profile in low-dimensional space. The mapper module has four fully connected layers with 16, 16, 16, and six neurons and uses LeakyReLU as the activation function. This module takes the output of the encoder module (i.e., latent representation of the joint profile) and targets the joint profile's AOI as input. It generates a latent representation of the target joint profile in output, which serves as input for the decoder module. The decoder module has five sets of one convolutional layer followed by one upsampling layer subsequently followed by one max-pooling layer. In addition, it has one more convolution layer, which is also the last layer of MoveNet. This layer also applies a linear activation function on the output. Each convolution layer uses a filter of size 3, and each upsampling layer uses a filter of size 2. Table I presents other relevant details of the architecture.

C. Training of MoveNet

There are two important features of the data set for a training point of view: richness in intrasubject versatility and poorness in intersubject versatility. Since the total numbers of samples in data matrices for phase I and phase II are 27 106 and 26 559, respectively, an average number of samples for a subject is around 2683 [((27 106 + 26 559)/2)/10 = 2683.25] for each phase. These 2683 samples have 27 different combinations of walking speed and AOI of the treadmill. However, the versatility in terms of subjects is poor as they are only ten in numbers. This implies that the data are incoming from ten different distributions. This is not a big problem until the NN only learns the significant features of the problem, which does not always happen during training an NN, as there are two ways by which the network can get high accuracy: one is by learning the task-relevant features (signal), and the other is by learning the task-irrelevant features (noise). Cross-validation techniques, such as k-fold cross validation, Monte Carlo cross validation, and the Holdout method, are unable to capture the learning of noise by NN as, in these methods, both test and training data are originated from the same population. On the other hand, leave-one-out cross validation can be used for this problem as it divides data into test and training according to the population distribution. Thus, the authors have used leave-one-out cross validation for all subjects (i.e., tenfold) and averaged the performance parameter's values across the folds. The same initial random weights of the network are used for each fold to provide the same initial random states across the folds. The motivation behind dividing the training into two phases is threefold.

1) We hypothesized that joint profiles for each subject and each condition have its position in the latent space of the autoencoder. These positions can be translated to a different position in the same latent space according to the input target AOI. This new position can be decoded by the decoder to generate the target joint profile. To test our hypothesis, we divided the training into two parts and trained the modules of the network on different data matrices.
2) We have compared the performance of the network when it is trained in two phases with when it is trained in a single phase (all three modules together).
3) Dividing the training into two phases not only gives better results but also provides ease in its implementation. With the help of transfer learning, the pretrained weights of the first phase (encoder and decoder module) trained on a large population can directly be used for learning weights of the second phase (mapper module) on a small population.

For any fold, nine subjects' data are divided into training (80%) and validation sets (20%), and data of the remaining one subject are treated as the testing set. The training set is used for training the network, the validation set is used for avoiding overtraining of the network, and the testing set is used for testing the performance of the trained network.

1) Phase I: A convolutional autoencoder (encoder + decoder modules) is trained and validated on the joint profile of nine subjects and tested on the remaining one subject. Here, all combinations of speed and AOI are considered for training so that the autoencoder could learn the actual representation of the joint profile. The data matrices are constructed and loaded to a python runtime, and each data point is standardized to zero mean and one SD. The standardization parameters, namely, mean, SD, and data matrices, are saved for the future reconstruction of the original signal from standardized data. A total of 140 epochs with a batch size of 16 samples are used during the training as they are found better than the other combinations. A model checkpoint of early stopping with 30 sample points on validation loss was used to avoid overfitting.

2) Phase II: The mapper module is trained and validated on the same nine subjects and tested on the same one subject

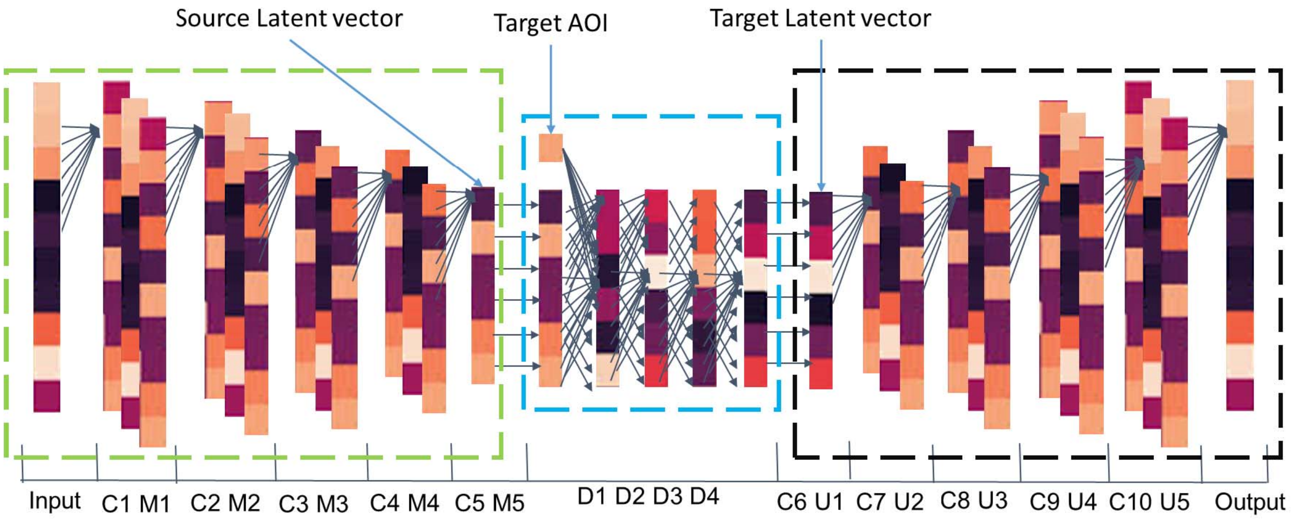

Fig. 2. Pictorial representation of the architecture of the proposed model: MoveNet. The networks enclosed in light green, blue, and black dotted boxes are the architectures of the encoder, mapper, and decoder modules, respectively. The abbreviations used as labels of the layers are denoting the corresponding layer's name, and details of these layers are presented in Table I.

TABLE I
ARCHITECTURE OF MOVENET

S.No	Layer Name	Layer Type	Filter size	Layer parameters	Stride	Output size	Parameters
1	Cl	Convolution 1	3	64 filters	1	192×64	256
2	M1	Max pooling 1	2	-	-	96×64	-
3	C2	Convolution 2	3	32 filters	1	96×32	6176
4	M2	Max pooling 2	2	-	-	48×32	-
5	C3	Convolution 3	3	16 filters	1	48×16	1552
6	M3	Max pooling 3	2	-	-	24×16	-
7	C4	Convolution 4	3	8 filters	1	24×8	392
8	M4	Max pooling 4	2	-	-	12×8	-
9	C5	Convolution 5	3	1 filter	1	12×1	25
10	M5	Max pooling 5	2	-	-	6×1	-
11	D1	Dense 1	-	16 neurons	-	16	128
12	D2	Dense 2	-	16 neurons	-	16	272
13	D3	Dense 3	-	16 neurons	-	16	272
14	D4	Dense 4	-	6 neurons	-	6	102
15	C6	Convolution 6	3	1 filter	1	6×1	4
16	U1	Up sampling 1	2	-	-	12×1	-
17	C7	Convolution 7	3	8 filters	1	12×8	32
18	U2	Up sampling 2	2	-	-	24×8	-
19	C8	Convolution 8	3	16 filters	1	24×16	400
20	U3	Up sampling 3	2	-	-	48×16	-
21	C9	Convolution 9	3	32 filters	1	48×32	1568
22	U4	Up sampling 4	2	-	-	96×32	-
23	C10	Convolution 10	3	64 filters	1	96×64	6208
24	U5	Up sampling 5	2	-	-	192×64	-
25	C11	Convolution 11	3	1 filter	1	192×1	193

as used in phase I. The data matrices are constructed and loaded to python runtime from the data set. Here, the input data contain a joint profile at 0° AOI at a constant speed and a float value for the target joint profile's AOI. The output data contain the target joint profile at the given target joint profile's AOI and at the same speed as the speed of input data. The target joint profile's AOI is rescaled by dividing with 10 to maintain its range between -1 and 1. For better learning and fast convergence of the network, the averaged target joint profile of the given AOI with given walking speed as the output data is considered. For computing these average target joint profiles for subjects, all samples corresponding to the target joint profile at a given walking speed are averaged to produce a single joint profile for each condition. MoveNet is trained to predict these averaged joint profiles. This process reduced the noise of the data and, thus, provided better learning. Now, each data point of both the data matrices (input and output) is standardized to zero mean and one SD. The standardization parameters and data matrices are saved as python variables using pickle library [39]. A total of 50 epochs with a batch size of 16 samples are used during the training as they are found better than the other combinations. A model checkpoint

TABLE II

SPECIFICATION OF THE SYSTEM

Name	Parameter
CPU RAM	13 GB
GPU RAM	16 GB
CPU	Intel(R) Xeon(R)@ 2.30GHz
Graphics processor	Tesla P100-PCIE
Cuda cores	2560
Memory type	GDDR5X
Language	Python 3.8.1

of early stopping with 30 sample points on validation loss is used to avoid overfitting. During this phase of the training, the weights of the encoder and decoder modules are kept frozen. For finding generation error, the input and the generated data matrices are destandardized.

D. Evaluation Metrics

The performance of a predictive NN algorithm is often evaluated by the difference between the values of ground-truth trajectory and the values of generated trajectory obtained from the NN. In this study, the mean absolute error (MAE) is used for quantifying the predictive performance of the model.

Let y' be the output trajectory generated from the NN for a given sample X, and Y' is the destandardized value of y'. Then, the absolute error for a gait cycle instance i of a data point is calculated, as shown in the following equation:

$$\mathrm{AE}_i = \left|Y_i' - Y_i\right| \tag{1}$$

where $Y = \{Y_i \mid i \leq 150, i \epsilon N\}$ is the recorded joint profile (ground truth) for a given sample X, and $Y' = \{Y_i' \mid i \leq 150, i \epsilon N\}$ is the destandardized output joint profile generated by the NN for a given sample X.

MAE for a data point is calculated, as shown in (2). It is measured in degree

$$\mathrm{MAE} = \frac{1}{150}\sum_{i=1}^{150}\mathrm{AE}_i. \tag{2}$$

The MAE roughly represents the generation error MoveNet produced for any gait cycle instance. For example, an MAE of 3° indicates that, for predicting joint angle at any time instance, MoveNet can produce ±3° error.

Since most of the previous studies used root mean squared error (RMSE) as evaluation metric, RMSE for MoveNet is also calculated for comparing its performance with previous studies. RMSE of MoveNet is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{150}\sum_{i=1}^{150}\left(Y_i' - Y_i\right)^2}. \tag{3}$$

III. RESULTS

A. Environment Settings

The NNs are trained and tested using the Keras framework and the Google Colab platform. The specification of the system is noted in Table II.

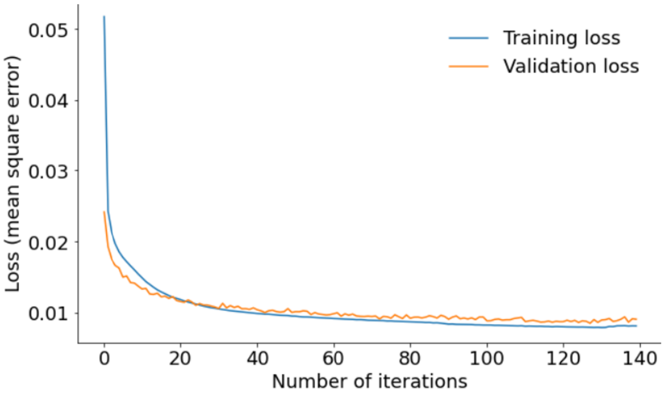

Fig. 3. Training loss and validation loss during I phase of the training.

TABLE III

MEAN (DEGREE) AND SD OF DISTRIBUTION OF SAMPLES' MAE FOR ALL AOIS (DEGREE). ALL WALKING SPEEDS (0.8, 1, AND 1.2 m/s) AND ALL FOLDS ARE CONSIDERED FOR COMPUTING ENTRIES OF THE TABLE

AOI	-10	-7.5	-5	-2.5	0	2.5	5	7.5	10
Mean	2.93	3.03	2.86	2.50	2.30	2.34	2.68	2.52	2.59
SD	1.48	1.47	1.28	1.04	0.95	0.94	1.05	1.04	1.00

For analysis, a system having the following specifications is used: Intel Core i5 6200U CPU @ 2.30 GHz, 4-GB DDR4 RAM, NVIDIA GeForce 930MX graphic engine, 64-bit Windows 10 Operating System, and MATLAB 2019a platform.

B. Training Parameters

The optimizer and the loss function used in both phases are the same. The loss function of MAE is used to update the weights as it is a direct measure of the generation error, and it is less prone to outliers than mean square error. The Adam optimizer with a learning rate of 0.001, $\beta 1 = 0.9$, $\beta 2 = 0.999$, and epsilon $= 10^7$ is used to update the weight of the network and find the local minima of the training loss function. $\beta 1$ and $\beta 2$ are the exponential decay rate of the first- and second-moment estimates of the Adam optimizer algorithm, respectively [40]. How training loss and validation loss vary with the number of iterations during phase I and phase II is shown in Figs. 3 and 4, respectively. These curves are calculated by averaging over the folds. In the figure, the X-axis denotes the number of iterations, which is also equivalent to training time. As the autoencoder is being trained, its performance is being increased, and its loss is being decreased. The Y-axis denotes the loss function's value, i.e., MAE of the samples for the given training epoch. Since the samples are standardized before feeding to the autoencoder network, the calculated loss has a different unit from degrees.

C. Testing Results

The performance of the model is calculated by testing it on test samples for each fold and then averaged across the

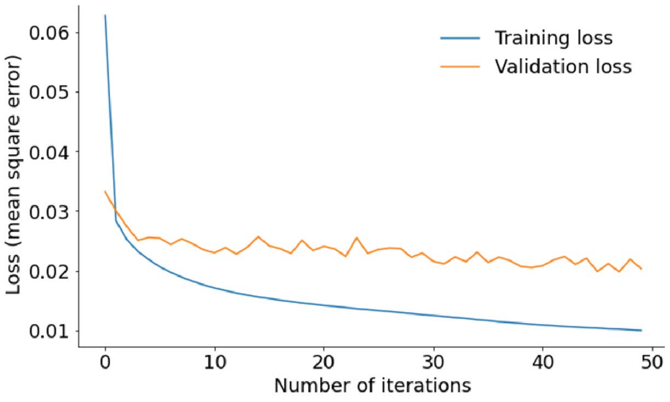

Fig. 4. Training loss and validation loss during II phase of the training.

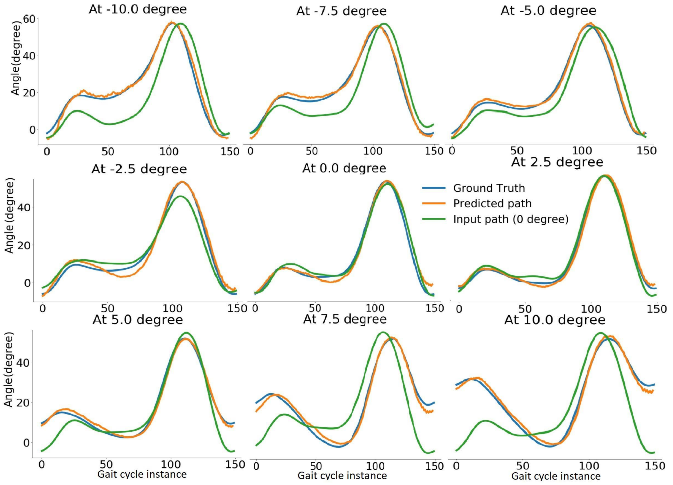

Fig. 5. Sample plots of ground truth (blue), predicted joint profile (orange), and input joint profile (green) for a subject at different target AOIs. Input joint profiles are gait cycles at 0° AOI. There is an intersubplot variability in the input joint profiles as different gait cycles are considered for different subplots. Ground truth is the average of gait cycles of the target AOI at the given walking speed and subject. The predicted path is the gait cycle generated by MoveNet from the given input joint profile and target AOI.

TABLE IV

Mean (Degree) and SD of Distribution of Samples' MAE for All Folds. All Walking Speeds (0.8, 1, and 1.2 m/s) and All AOIs Are Considered for Computing Entries of the Table

Fold	1	2	3	4	5	6	7	8	9	10
Mean	2.87	3.06	3.67	2.09	2.08	2.64	3.42	2.79	1.98	1.98
SD	0.97	1.13	1.36	0.88	0.57	1.12	1.48	1.00	0.89	0.57

TABLE V

Mean (Degree) and SD of Distribution of Samples' MAE for All Walking Speeds. All AOIs and All Folds Are Considered for Computing Entries of the Table

Speed (m/s)	0.8	1	1.2
Mean	2.69	2.60	2.63
SD	1.25	1.15	1.15

folds. The network obtained RMSE of 3.24 ± 1.19° and MAE of 2.66 ± 1.00° on the test samples. The sample outputs of the model on nine test samples with different target AOIs are presented in Fig. 5. The figure suggests that the model is able to predict the target joint profiles with acceptable generation error. A better insight into the dependence of the model's performance on the target AOI is provided in Fig. 6 and Table III. This figure shows the distribution of the generation error (i.e., MAE) calculated for each test sample at different target AOIs. One-way Analysis of Variance (ANOVA) with the Bonferroni correction is used for testing the significance of the difference in means of the distributions. During this, multiple-comparison results with a p-value lesser than 0.05 are considered significant. These results suggest that the model's performance is excellent for all AOIs as the mean of the generation errors lies in a range of 2.30°–3.03°. However, there is a significant difference in the mean of the distribution of MAE across the AOI's. This difference suggests that the model is performing slightly better for small AOIs than large AOIs. A similar analysis is performed for folds, which is shown in Fig. 7 and Table IV. The results suggest that the model's performance is excellent for all folds as means of the generation error lie in a range of 1.98°–3.67°, which also means that the performance of the model is remarkable for all the subjects. Fig. 8 and Table V depict the distribution of MAE for test samples at different walking speeds. The mean of generation is in a range of 2.60°–2.69°; this suggests that the model is performing slightly worse at the 0.8-m/s walking

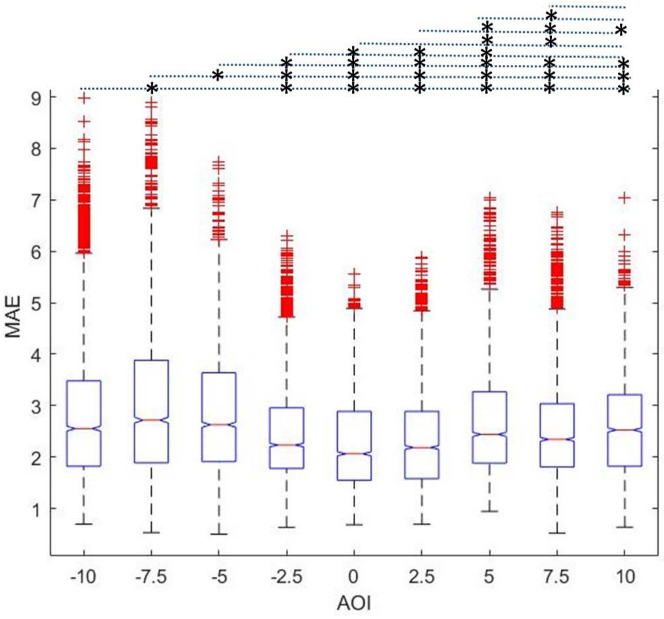

Fig. 6. Distribution of MAE of test samples for each AOI: test samples are divided into nine groups according to their target joint profile's AOI. Then, MAE is calculated for each test sample and each group. The distribution MAE for each group is shown in the above figure. Here, the red line in each box represents the median of the distribution of MAE in the corresponding group. Outliers are shown with red plus marks for each group. The round dotted line (black color) and asterisk marks drawn on the top of the figure depict a significant difference in the mean values of the MAE distribution of a group with other groups. For example, the bottommost line and asterisk marks show a significant difference of the mean values of the MAE distribution of group 1 (i.e., −10° AOI) with group 2, group 4, group 5, group 6, group 7, group 8, and group 9. Data from all walking speeds (0.8, 1, and 1.2 m/s) and all folds are shown in this plot. The number of outliers is 1% of the number of samples in each group.

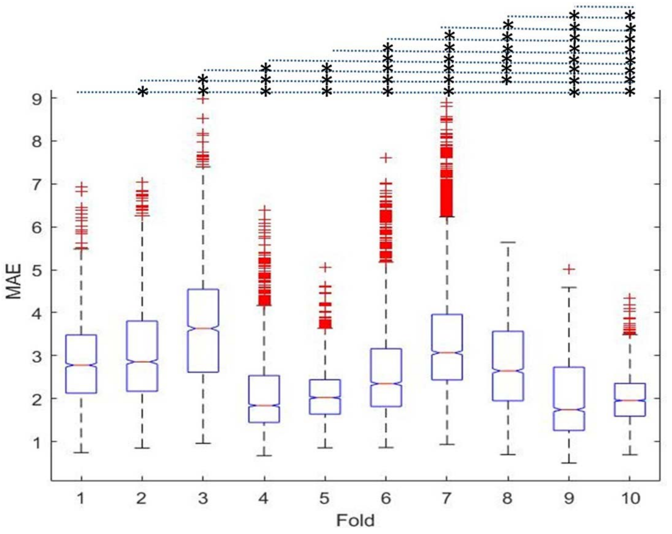

Fig. 7. Distribution of MAE of test samples for each fold: since leave-one-out cross validation is used during the training and testing of the model, the methodology has tenfold. In each fold, the training set comprises nine subjects, and the testing set comprises one subject. The performance of the model for each fold (group) in terms of the distribution of MAE of its testing samples. Here, the red line in each box represents the median of the distribution of MAE in the corresponding group. Outliers are shown with red plus marks for each group. The round dotted line (black color) and asterisk marks drawn on the top of the figure depict a significant difference in the mean values of the MAE distribution of a group with other groups. For example, the topmost line with no asterisk marks shows the difference of the mean values of the MAE distribution of fold 9 with fold 10 is not significant. Data from all walking speeds (0.8, 1, and 1.2 m/s) and all AOIs are shown in this plot. The number of outliers is 1% of the number of samples in each group.

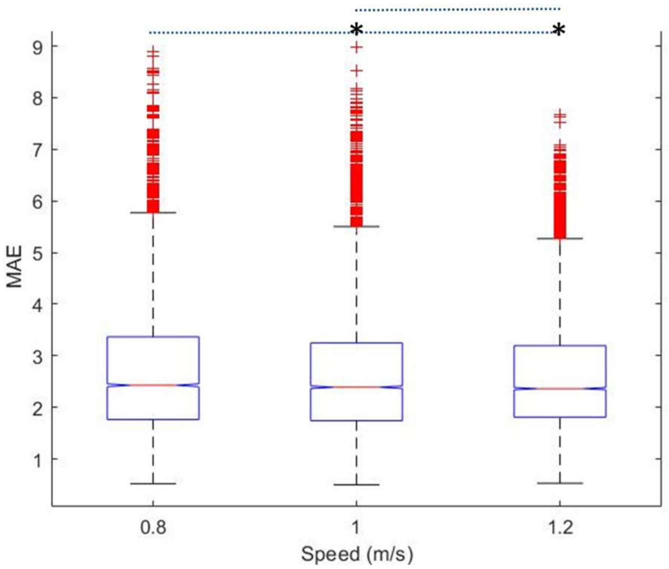

Fig. 8. Distribution of MAE of test samples for each speed: the model is trained and tested for three walking speeds. The performance of the model for each speed (group) in terms of the distribution of MAE of its testing samples. Here, the red line in each box represents the median of the distribution of MAE in the corresponding group. Outliers are shown with red plus marks for each group. The round dotted line (black color) and asterisk marks drawn on the top of the figure depict a significant difference in the mean values of the MAE distribution of a group with other groups. For example, the topmost line with no asterisk marks shows the difference of the mean values of the MAE distribution of 1-m/s speed with 1.2-m/s speed being not significant. Data from all folds and all AOIs are shown in this plot. The number of outliers is 1% of the number of samples in each group.

speed than other walking speeds. Furthermore, the mean and SD of MAE of all folds, AOIs, and walking speeds are also computed. The mean of 2.66° and the SD of generation error of 1.00° at any gait cycle instance show that MoveNet is able to predict reference joint profile at a given AOI with excellent accuracy and precision, respectively.

These results validate the authors' hypothesis. The performance of MoveNet in phase I suggests that joint profiles for each subject and each condition have their position in the autoencoder's latent space. Since the weights of the encoder and the decoder are frozen during phase II, the latent space of both encoder and decoder is the same. The performance of MoveNet in phase II suggested that the mapper can lean to map low-dimensional representation of the source trajectory to low-dimensional representation of the target trajectory.

D. Ablation Study

The architecture of MoveNet is decided after performing ablation studies. In the first study, all three modules of the network are trained simultaneously. The comparison of the results of this network with MoveNet is shown in Table VI. From the results, one can infer that dividing the training into two parts improves the learning of the network. In the second study, the number of convolutional layers and batch normalization layers of encoder and decoder modules on the performance of MoveNet is assessed. These results are presented in Table VII. In the third study, the number of dense layers of mapper modules on the performance of MoveNet is assessed. These results are presented in Table VIII. In the fourth study, the effect of varying sample sizes on the performance of MoveNet is assessed. These results are presented in Table IX. The previous studies consider RMES value less than 8° as an acceptable error for an exoskeleton [21], [41]. The fourth study's results suggest that only one subject is sufficient for training MoveNet to predict joint angle profiles of the remaining nine subjects with an acceptable error.

TABLE VI

Ablation Study 1: Mean (Degree) and SD (Degree) of Test Samples' MAE When the Modules Are Trained Simultaneously and When the Modules Trained Separately

Training of the modules	Simultaneously	Separately
Mean	2.82	2.66
SD	0.97	1.00

TABLE VII

Ablation Study 2: Mean (Degree) and SD (Degree) of Test Samples' MAE for MoveNet With Variable Number of Convolutional Layers and Batch Normalization Layers of Encoder and Decoder Modules

Number of layers	1	2	3	4	5
Mean	27.15	28.54	7.12	3.98	2.66
SD	7.38	5.40	2.44	1.69	1.00

E. Performance of MoveNet on Unseen AOIs

To test the capability of MoveNet in predicting the unseen AOIs, a separate study is performed. Here, MoveNet is trained

TABLE VIII

Ablation Study 3: Mean (Degree) and SD (Degree) of Test Samples' MAE for MoveNet With Variable Number of Layers of Mapper Modules

Number of layers	1	2	3	4
Mean	10.51	5.24	3.28	2.66
SD	4.51	1.89	1.22	1.00

TABLE IX

Ablation Study 4: Mean (Degree) and SD (Degree) of Test Samples' MAE for MoveNet With Variable Number of Subjects/Sample Size (#Sub)

#Sub	1	2	3	4	5	6	7	8	9	10
Mean	3.66	3.45	3.26	3.12	2.98	2.69	2.70	2.69	2.67	2.66
SD	1.25	1.16	1.12	1.23	1.31	1.32	1.25	1.14	1.15	1.00

TABLE X

Mean (Degree) and SD (Degree) of Distribution of Samples' MAE for All AOIs (Degree). All Walking Speeds (0.8, 1, and 1.2 m/s) and All Folds Are Considered for Computing Entries of the Table. Bold Text Indicates Unseen AOIs

AOI	-10	**-7.5**	-5	**-2.5**	0	**2.5**	5	**7.5**	10
Mean	2.38	3.32	2.52	2.83	2.53	2.56	2.59	2.97	2.69
SD	0.94	1.54	1.20	1.29	0.96	1.49	0.82	1.56	0.91

on −10°, −5°, 0°, 5°, and 10° and tested on −7.5°, −2.5°, 2.5°, and 7.5° AOIs. These results are shown in Table X. These results suggest that MoveNet is capable of predicting joint angle profiles at the unseen AOIs within a range with excellent accuracy.

F. Correlation Analysis

A correlation analysis is performed to verify the nonredundancy of information and independence of the extracted parameters. After training phase I, all samples of the input data matrix are feedforwarded to the autoencoder. The low-dimensional representation of each data point of the input data matrix is stored in a vector of length six. Here, each index of the vector corresponds to a node of the bottleneck of the autoencoder, i.e., the output of the encoder module (see Fig. 2). By considering the vectors of all data points, a correlation analysis is performed between the values of these six nodes. The correlation coefficient, the linear fit, and the scatter plot for each node pair are shown in Fig. 9. These results suggest that all six nodes represent different properties of the gait profile. The distribution of values of each node is also shown in Fig. 9. The broad range of values of nodes shown in the histogram indicates that each node is representing gait-relevant information.

IV. Discussion

The kinematic events of the gait cycle are generally processed in the time domain, where each time point corresponds to a specific event of the gait cycle. There are two essential properties of these events for the artificial generation of gait cycles; they are predictable, and they are correlated with each other. However, it is also considered that the kinematics of gait is user-dependent [35]. It suggests that both gait cycle-specific and user-specific parameters are required to predict user-specific gait kinematics. However, defining these parameters is not a trivial task. Therefore, an NN for extracting these parameters from raw data of joint angle profile is used in this study. The outcomes of the work indicate the potential of the present approach in minimizing the measurement time by avoiding the measurement at different slopes. In this work, the estimation of joint angles at different slopes is done using only a computation approach and was observed to agree with experimentally recorded data.

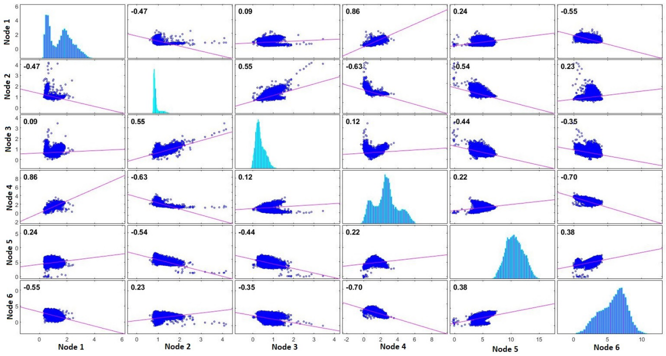

Fig. 9. Scatter plots and histograms of the values of nodes of the bottleneck of the autoencoder for all samples: each diagonal subplot of the figures shows the histogram of values of the corresponding node/neuron. Each nondiagonal subplot shows a scatter plot, fit line, and correlation coefficient of values of a pair of the nodes. The correlation coefficient is shown on the top left of the subplots.

The performance of the proposed model is compared with seven state-of-the-art methods. The comparison results are presented in tabular form in Table XI. However, according to the best of our knowledge, there is no existing work in the literature that predicted multiple joint angle profiles at various slopes from a single-input joint angle profile. For computing time-efficiency of MoveNet, the system specification shown in Table II is considered. Cunha *et al.* [22], Liang *et al.* [24], Ferreira *et al.* [27], and Ren *et al.* [33] used artificial neural network for predicting joint angle profiles. The results of trained models introduced by Cunha *et al.* [22], Liang *et al.* [24], Ferreira *et al.* [27], and Ren *et al.* [33] are prone to overfitting as they trained and tested their models on the same population. Thus, the implementation of these models in an exoskeleton working in real-life scenarios is questionable. On the other hand, the reported results of the testing of MoveNet are calculated by considering training and testing data from a different population. Fukuchi and Duarte [28] exploited a second-order polynomial for fitting angle profile curve. Their approach is fast as a smaller number of computations are needed by a second-order polynomial for prediction. However, their approach is less accurate as the angle profile follows a complex curve different from the second-order polynomial. The model proposed by Moissenet *et al.* [30] is trained and tested on the same values of SSP. This limits the use of the model in real-life scenarios where unseen values of SSPs exit. He *et al.* [23] and

TABLE XI

Comparison With Previous Works on Knee Joint Angle Generation (RMSE in °) for Able-Bodied Subjects

Source	Method	Input	RMSE (in deg)	Time for one sample (s)
[22]^a	BNN	Height, weight, age and gait speed	3.28	-
[28]^a	Second-order polynomial	Gait speed	4.45	-
[30]^a	Multiple regression	Gait speed, gender, age, BMI	4.78	-
[33]^b	Random Forest	SSP	2.72	2.5422
[23]^a	ELM+Autoencoder	SSP and joint profiles	2.48	-
[24]^a	LSTM	Joint profiles	2.23	0.0852
[27]^a	ELM	Height, weight, age and gait speed	2.51	0.0100
Proposed ^{c}	Autoencoder+dense NN	Joint profile	3.24	0.0052

Usability: ^{a} Multiple gait speeds, ^{b} Single gait speed, ^{c} Multiple gait speeds and multiple slopes

Liang *et al.* [24] are relatively slower methods and, thus, cannot be used for real-time applications. The data required as an input of the model proposed in [24] are very large as it needs 21 SSPs for angle profile prediction. Recording such many parameters for gait prediction can take an ample amount of time and effort. On the other hand, MoveNet needs a joint angle profile of one cycle as an input. The implementation of MoveNet is easier than most of the previous methods as the architecture of MoveNet can easily be developed with a deep learning framework, such as Keras, PyTorch, and TensorFlow. The training and testing times of MoveNet vary with the data size and system configuration. When trained on ten subjects with 26 599 number of samples, MoveNet takes 21.6 min. The results shown in Table XI show that the prediction time of one sample by MoveNet is 0.0052. Thus, it can be used for real-time applications. MoveNet attained an excellent generation performance and showed its potentials for using it with future generation prosthesis and exoskeletons. Since it can produce a joint angle profile at any instant of the gait cycle for continuously varying speeds and slopes, it can reduce control-related computations of the controller of an exoskeleton. The authors' next work will be primarily focused on the implementation of MoveNet in a powered exoskeleton. However, there is one limitation of MoveNet that limits its use to only abled-body users. MoveNet needs a sample joint angle profile recorded from the able-bodied users. For a differently abled person, it can be challenging to record the required sample joint profile. Thus, in the future, the authors will try to develop an intelligent system, which can predict user-specific joint profiles for differently abled persons that can be used for rehabilitation purposes.

V. Conclusion

In conclusion, the study presents a deep NN for knee joint profile prediction at variable walking speeds and slopes for able-bodied users. This proposed model is trained and verified on a public data set having data of 270 min recorded from ten subjects. MAE of 2.66 ± 1.00° obtained from leave-one-out cross validation indicated that MoveNet can accurately predict angle profiles for an unfamiliar user. This study's key contribution is to present a method that can predict highly user-specific profiles from minimum input data. This contribution has dual benefits. First, it can reduce the gait data recording time as data recorded from the flat surface can predict gait data for other AOIs. Second, it can propose continuous reference trajectories to an exoskeleton during variable walking speeds and slopes.

Acknowledgment

The authors thank a research team from The University of Texas at Dallas, particularly Kyle R. Embry, Dario J. Villarreal, Rebecca L. Macaluso, and Robert D. Gregg, for making the used data set publicly available.

References

[1] A. H. Dewolf, Y. Ivanenko, K. E. Zelik, F. Lacquaniti, and P. A. Willems, "Kinematic patterns while walking on a slope at different speeds," *J. Appl. Physiol.*, vol. 125, no. 2, pp. 642–653, Aug. 2018.

[2] L.-Y. Guo *et al.*, "Effects of speed and incline on lower extremity kinematics during treadmill jogging in healthy subjects," *Biomed. Eng., Appl., Basis Commun.*, vol. 18, no. 2, pp. 73–79, Apr. 2006.

[3] L. Biedermann, W. Matthis, and C. Schulz, "Leg prosthesis with an artificial knee joint and method for controlling a leg prosthesis," U.S. Patent 6 755 870, Jun. 29, 2004.

[4] D. May, "Artificial leg with stable link-type knee joint," U.S. Patent 3 823 424, Jul. 16, 1974.

[5] H. Rïfai, S. Mohammed, K. Djouani, and Y. Amirat, "Toward lower limbs functional rehabilitation through a knee-joint exoskeleton," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 2, pp. 712–719, Mar. 2017.

[6] D. Wang, K.-M. Lee, J. Guo, and C.-J. Yang, "Adaptive knee joint exoskeleton based on biological geometries," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1268–1278, Aug. 2014.

[7] R. P. Hirata and M. Duarte, "Effect of relative knee position on internal mechanical loading during squatting," *Brazilian J. Phys. Therapy*, vol. 11, no. 2, pp. 121–125, 2007.

[8] M. Grimmer, M. Eslamy, and A. Seyfarth, "Energetic and peak power advantages of series elastic actuators in an actuated prosthetic leg for walking and running," *Actuators*, vol. 3, no. 1, pp. 1–19, 2014.

[9] H.-G. Kim, S. Park, and C. Han, "Design of a novel knee joint for an exoskeleton with good energy efficiency for load-carrying augmentation," *J. Mech. Sci. Technol.*, vol. 28, no. 11, pp. 4361–4367, Nov. 2014.

[10] B. Laschowski, J. McPhee, and J. Andrysek, "Lower-limb prostheses and exoskeletons with energy regeneration: Mechatronic design and optimization review," *J. Mech. Robot.*, vol. 11, no. 4, Aug. 2019, Art. no. 040801.

[11] J. Chen, X. Zhang, Y. Cheng, and N. Xi, "Surface EMG based continuous estimation of human lower limb joint angles by using deep belief networks," *Biomed. Signal Process. Control*, vol. 40, pp. 335–342, Feb. 2018.

[12] K. Liu, T. Liu, K. Shibata, and Y. Inoue, "Ambulatory measurement and analysis of the lower limb 3D posture using wearable sensor system," in *Proc. Int. Conf. Mechatron. Automat.*, Aug. 2009, pp. 3065–3069.

[13] L. Kun, Y. Inoue, K. Shibata, and C. Enguo, "Ambulatory estimation of knee-joint kinematics in anatomical coordinate system using accelerometers and magnetometers," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 2, pp. 435–442, Feb. 2011.

[14] N. Mezghani *et al.*, "Automatic classification of asymptomatic and osteoarthritis knee gait patterns using kinematic data features and the nearest neighbor classifier," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 3, pp. 1230–1232, Mar. 2008.

[15] K. R. Embry, D. J. Villarreal, and R. D. Gregg, “A unified parameterization of human gait across ambulation modes,” in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 2179–2183.

[16] K. R. Embry, D. J. Villarreal, R. L. Macaluso, and R. D. Gregg, “Modeling the kinematics of human locomotion over continuously varying speeds and inclines,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 12, pp. 2342–2350, Dec. 2018.

[17] A. K. Godiyal and D. Joshi, “Optimal force myography placement for maximizing locomotion classification accuracy in transfemoral amputees: A pilot study,” *IEEE J. Biomed. Health Informat.*, vol. 25, no. 4, pp. 959–968, Apr. 2021.

[18] G. Cooper *et al.*, “Inertial sensor-based knee flexion/extension angle estimation,” *J. Biomechanics*, vol. 42, no. 16, pp. 2678–2685, Dec. 2009.

[19] C. Jakob *et al.*, “Estimation of the knee flexion-extension angle during dynamic sport motions using body-worn inertial sensors,” in *Proc. 8th Int. Conf. Body Area Netw.*, 2013, pp. 289–295.

[20] A. Ancillao, S. Rossi, and P. Cappa, “Analysis of knee strength measurements performed by a hand-held multicomponent dynamometer and optoelectronic system,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 1, pp. 85–92, Jan. 2017.

[21] A. Kumar, A. K. Godiyal, P. Joshi, and D. Joshi, “A new force myography-based approach for continuous estimation of knee joint angle in lower limb amputees and able-bodied subjects,” *IEEE J. Biomed. Health Informat.*, vol. 25, no. 3, pp. 701–710, Mar. 2021.

[22] P. S. Cunha, J. Ferreira, A. P. Coimbra, and M. Crisóstomo, “Computational intelligence generation of subject-specific knee and hip healthy joint angles reference curves,” in *Proc. Medit. Conf. Med. Biol. Eng. Comput.* Coimbra, Portugal: Springer, 2019, pp. 1653–1668.

[23] Y. He *et al.*, “GC-IGTG: A rehabilitation gait trajectory generation algorithm for lower extremity exoskeleton,” in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 2031–2036.

[24] F.-Y. Liang *et al.*, “Online adaptive and LSTM-based trajectory generation of lower limb exoskeletons for stroke rehabilitation,” in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 27–32.

[25] D.-X. Liu, X. Wu, C. Wang, and C. Chen, “Gait trajectory prediction for lower-limb exoskeleton based on deep spatial-temporal model (DSTM),” in *Proc. 2nd Int. Conf. Adv. Robot. Mechatron. (ICARM)*, Aug. 2017, pp. 564–569.

[26] M. Błażkiewicz and A. Wit, “Artificial neural network simulation of lower limb joint angles in normal and impaired human gait,” *Acta Bioeng. Biomech.*, vol. 20, no. 4, pp. 43–49, 2018.

[27] J. P. Ferreira, A. Vieira, P. Ferreira, M. Crisóstomo, and A. P. Coimbra, “Human knee joint walking pattern generation using computational intelligence techniques,” *Neural Comput. Appl.*, vol. 30, no. 6, pp. 1701–1713, Sep. 2018.

[28] C. A. Fukuchi and M. Duarte, “A prediction method of speed-dependent walking patterns for healthy individuals,” *Gait Posture*, vol. 68, pp. 280–284, Feb. 2019.

[29] S. A. Ali, K. A. M. Annuar, and M. F. Miskon, “Trajectory planning for exoskeleton robot by using cubic and quintic polynomial equation,” *Int. J. Appl. Eng. Res.*, vol. 11, no. 13, pp. 7943–7946, 2016.

[30] F. Moissenet, F. Leboeuf, and S. Armand, “Lower limb sagittal gait kinematics can be predicted based on walking speed, gender, age and BMI,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Dec. 2019.

[31] J. Hong, C. Chun, S.-J. Kim, and F. C. Park, “Gaussian process trajectory learning and synthesis of individualized gait motions,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 6, pp. 1236–1245, Jun. 2019.

[32] A. J. J. Smith, E. D. Lemaire, and J. Nantel, “Lower limb sagittal kinematic and kinetic modeling of very slow walking for gait trajectory scaling,” *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0203934.

[33] S. Ren *et al.*, “Personalized gait trajectory generation based on anthropometric features using random forest,” *J. Ambient Intell. Hum. Comput.*, pp. 1–12, Jul. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s12652-019-01390-3#citeas

[34] T. Bao, S. A. R. Zaidi, S. Xie, P. Yang, and Z.-Q. Zhang, “A CNN-LSTM hybrid model for wrist kinematics estimation using surface electromyography,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[35] F. Horst, S. Lapuschkin, W. Samek, K.-R. Müller, and W. I. Schöllhorn, “Explaining the unique nature of individual gait patterns with deep learning,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–13, Dec. 2019.

[36] K. Embry, D. Villarreal, R. Macaluso, and R. Gregg, “The effect of walking incline and speed on human leg kinematics, kinetics, and EMG,” Tech. Rep., 2018. [Online]. Available: https://ieee-dataport.org/open-access/effect-walking-incline-and-speed-human-leg-kinematics-kinetics-and-emg, doi: 10.21227/gk32-e868.

[37] M. E. Hahn and K. B. O’Keefe, “A neural network model for estimation of net joint moments during normal gait,” *J. Musculoskeletal Res.*, vol. 11, no. 03, pp. 117–126, Sep. 2008.

[38] J. P. Ferreira, M. M. Crisostomo, and A. P. Coimbra, “Human gait acquisition and characterization,” *IEEE Trans. Instrum. Meas.*, vol. 58, no. 9, pp. 2979–2988, Sep. 2009.

[39] G. Van Rossum, “The Python library reference, release 3.8.2,” Python Softw. Found., Wilmington, DE, USA, Tech. Rep., 2020. [Online]. Available: http://citebay.com/how-to-cite/python-pickle/#:~:text=Van%20Rossum%20G., Python%20Software%20Foundation%3B%202020

[40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[41] Q. Meng *et al.*, “Pilot study of a powered exoskeleton for upper limb rehabilitation based on the wheelchair,” *BioMed Res. Int.*, vol. 2019, pp. 1–11, Dec. 2019.

Rishabh Bajpai received the B.Tech. degree in mechanical engineering from the PDPM Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Jabalpur, India, in 2020. He is currently pursuing the Ph.D. degree with the Centre for Biomedical Engineering, IIT Delhi, New Delhi, India, and the All India Institute of Medical Sciences (AIIMS) Delhi, New Delhi.

He has been working in the area of brain–computer interface, rehabilitation engineering, neuroscience, electroencephalography, and the development of biomedical instrumentation for applications specific to assistive devices for the disabled.

Deepak Joshi received the Ph.D. degree in biomedical engineering from IIT Delhi, New Delhi, India, in 2012.

He worked at various places, including the National University of Singapore, Singapore, Newcastle University, Newcastle upon Tyne, U.K., and the University of Oregon, Eugene, OR, USA. He is currently a Faculty Member in biomedical engineering with IIT Delhi and the All India Institute of Medical Sciences Delhi, New Delhi. He has been working in the areas of neuroprosthetics and neurorehabilitation for last nearly 14 years. His current research work combines experimental and computational techniques to understand the neural correlates during balancing and seamless transitions during walking. Besides that, he is also actively engaged in projects related to the development of wearable devices for applications specific to the diagnosis of neuromuscular disorders, assistive devices for the elderly and disabled, and biofeedback for rehabilitation in stroke patients.