

# Deep Learning-driven Front-Following within Close Proximity: a Hands-Free Control Model on a Smart Walker

Zhao Chongyu<sup>1</sup>, Guo Wenzhi<sup>2</sup>, Wen Rongwei<sup>3</sup> Zheng Wang, Chuan Wu

**Abstract**—With the ever-increasing elderly population, elder walking assistance is in strong demand. Instead of receiving assistance from a human carer, a smart walker can bring an elder user a more convenient and autonomous walking experience. Towards intelligent and safe walking assistance, we propose a close-proximity front-following model for smart walkers, which analyzes the walking gait and detects the walking intention of the user, and intelligently follows the user in the front to provide walking support, without the user pushing the walker. We design a deep learning model named Front-Following Net (FFLNet), consisting of CNN and LSTM networks to extract spatial and temporal features of the elder walking gait, collected in time windows through a thermal camera and a 2D LiDAR, for effective walking intention detection. As compared to other walking intention detection approaches, our model can explore more effective information in the gait data within a short walking period, and achieve accurate hands-free tracking of the user. Experiments show that our FFLNet can achieve over 77% detection accuracy among six representative walking intentions and more than 90% accuracy for turning intentions. Combined with a carefully designed walker control policy, our smart walker can achieve high front-following correctness with the user.

**Keywords:** Front Following, Smart Walker, Close Proximity, LSTM, Gait Analysis

## I. INTRODUCTION

Recent years have witnessed a significant rise in the elder population and a substantial increase in global life expectancy (from 66.8 years in 2000 to 73.3 years in 2019 [1]). Various research has exhibited the benefit of daily walking in improving elders' health and life situation [2]. Many elders are suffering from various mobility impairments.

A number of smart walkers have been designed for this problem [3][4][5][6]. Some [3][4] utilize sensors such as a joystick or pressure sensors for the user to move the walker; the user needs to actively provide operating signals for the walker to move along with him/her, which may lay burden on the older user. We aim to design a walker control model with better intelligence to automatically follow the user within close proximity in the front while the user is walking with the walker in a hands-free mode (see Fig. 1(A)). This walker

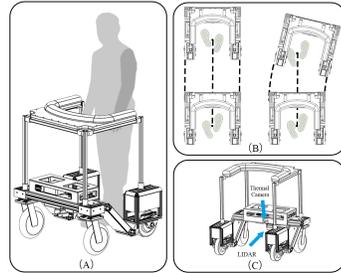


Fig. 1. (A) An elder person uses the walker in a hands-free manner. The walker can follow the user in close proximity to offer safety support. (B) Correct front following and tracking failure of the walker, respectively. (C) Overview of the smart walker and positions of a thermal camera and a 2D LiDAR.

can offer sufficient walking support while releasing the active operation burden for the user. The close proximity following brings challenges for walker control. As shown in Fig. 1(B), a collision between the user and the walker may well happen when the control model does not perform well. User position and walking intention detection is required for the control model to perform accurate and intelligent front-following.

Some smart walkers use laser or infrared sensors to detect the position of the lower limb of the user [7][8]. The measured distance between the user and the walker is used for adjusting the moving speed or braking of the walker [9]. Due to the low spatial resolution of these sensors, recognizing the pose of the user from the distance data typically requires a complex leg detection algorithm for leg separation [10]. After separation, the gait intention can be recognized by classifying limited location patterns in roughly divided regions [8]. However, without the feet orientation information, user walking intention may not be well detected. For example, with similar leg positions, the orientation between the two feet in cases of stepping forward and turning may well differ, revealing different walking intentions.

There exist smart walker designs which use other sensors (e.g., ultrasonic sensors, tri-axial load cells, RGB-D camera) to detect user walking intention [8]. Sierra *et al.* [5] use ultrasonic sensors to detect the presence of the user, and tri-axial load cells with adaptive filters (e.g., Karmen Filters) to estimate the walking gait. In [6], RGB-D camera data is fed to a convolution neural network (CNN)-based model to estimate human pose for gait estimation for walker control. These designs largely rely on instantaneous data to infer the walking intention, and seldom exploit temporal features of human walking in a consecutive walking process. Further, there is a lack of studies on robotic close-proximity front following designs with deep learning methods, combining temporal and spatial features.

<sup>1</sup>Zhao Chongyu, Department of Computer Science, the University of Hong Kong

<sup>2</sup>Guo Wenzhi, Department of the Mechanical Engineering, the University of Hong Kong

<sup>3</sup>Wen Rongwei, Department of Computer Science, the University of Hong Kong

Zheng Wang, Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, China

Chuan Wu, Department of Computer Science, the University of Hong Kong

This work was supported by Hong Kong Innovation and Technology Commission's Innovation and Technology Fund (Award No. ITS/305/19FP).

We develop a front-following model on a smart walker that utilizes the 2D LiDAR data and lower limb gait images. The 2D LiDAR detects the user position to maintain close-proximity and safety while the lower limb thermal gait images indicate the walking intention. We seek to collect consecutive lower limb images of the user through a non-intrusive thermal sensor. The basic design of our smart walker can be found in [11], together with a preliminary walker front-following model: a neural network with 2 fully-connection layers takes as input 8 gait images and leg position vectors in a small period of a clear human step, and produces a possible foot position, for inferring a movement intention in this step. With this design, the walker can only front-follow the user step by step, incurring jitters in its movement. In the new front-following model presented in this paper, we utilize more temporal and spatial features of the gait data. Instead of clearly defining a step of the user, we analyze all possible consecutive movements of the user. Main contributions of this paper are summarized as follow:

- We propose a close-proximity front-following control model, FFLNet, that can drive a walker to follow a user's walk smoothly, automatically and safely in the front.
- We utilize a thermal camera and a 2D LiDAR to detect the gait of the user for cost-effectiveness and better privacy protection.
- A combination of CNN, LSTM [12], and fully-connection layers is applied to carefully extract temporal and spatial gait information within time windows. Consecutive gait data and the last gait image in each time window are exploited separately in our model to achieve high accuracy in walking intention detection.
- Two gait datasets are collected for effective FFLNet learning including consecutive normal gait data of usual walking with the walker and the static gait data that seldom appears in the previous procedure. The two datasets enable the FFLNet extract features of all kinds of gait data.
- Experiments show that our proposed model can distinguish different walking intentions accurately, especially for the turning intentions the FFLNet precision can reach over 90%. The combined control policy can further improve the front-following action and achieve hands-free tracking with close proximity.

## II. RELATED WORK

### A. Robotic Following Models

In the domain of Human-Robot Interaction (HRI), robotics following remains an active research topic. In some designs, robots can detect, locate and follow the human user by sensors such as LiDAR or LRF to scan the environment and detect human legs for tracking [13][14][15][16]. Some robots use cameras and compute human positions with computer vision methods to achieve following [17][18]. In recent years, deep learning approaches for robotic following have gained attention. The robot in [19] adapts the MobileNet [20] to

obtain a bounding box of humans from 2D images, which is then converted to 3D locations for human detection and following. Similar designs of applying NNs and RGB-D images to detect human positions can be found in [21][22].

Compared to these designs, our approach utilizes 2D LiDAR and a low-resolution thermal camera. The 2D LiDAR can measure the user's position while the thermal camera can offer the lower-limb images that can be analyzed to obtain the walking intention. Though the LiDAR information can recognize the user walking intention to some level, we argue that the lower-limb images can offer more walking information before the user's position changes. Our design ensures users' privacy since we exploit low-resolution information from the thermal camera compared to the design that utilizes a high-resolution optical camera.

### B. Learning Human Gait

Human gait data are commonly used for analyzing human walking patterns for HRI. Some studies use lower limb data collected by LiDAR or LRF [10]. Some designs [23][24] adopt deep neural networks such as a CNN to achieve human identification based on image features. They extract walking image sequences from videos to obtain gait energy images [25] as the input to their deep learning models. Also, some researchers study abnormal gaits for disease diagnosis, e.g., Parkinson's disease, by detecting pressure changes of the insole [26] or accelerator readings [27].

The *FFLNet* in our design combines CNN for spatial features extraction and the LSTM [12] for temporal features extraction. Unlike the method in [27] that inputs all the data to an LSTM [12] within an observation period, we argue that to predict the walking intention from the user's gait data, the learning model should analyze the adjacent past gait information and the latest information separately since the latest frame represents the intention more. *FFLNet* adopts this separate analysis structure, which seldom appears in other learning models for gait prediction. This design achieves better performance.

## III. PROBLEM FORMULATION

### A. Walking Intention Detection Model

The proposed front-following model is designed on a smart walker that offers walking and fall support for an elder user in close proximity. In existing close-proximity following designs [5][6], sensors such as LRF keep scanning the user to detect leg positions and orientation changes. Few have exploited the feet angle, which can better represent the walking intention. With similar body positions, the walking intention can be different if the orientation of the feet is different. For example, when the user has just switched from standing still to turning to left on the same spot, the lower limb position is similar before and after, while the feet orientations differ.

We seek to estimate human walking intention  $\zeta$  according to the observed gait data including the feet orientations and leg positions. As human walking is a consecutive process, we derive the walking intention based on a time window  $T$  of

observations and define the walking intention detection function as  $\zeta_T = h(G_T, P_T)$ . Here  $G_T = \{g_1, g_2, \dots, g_n\}$  is the feet orientation observation sequence, containing  $n$  consecutive images captured by a thermal camera in the time window  $T$ .  $P_T = \{p_1, p_2, \dots, p_n\}$  is the leg position sequence, including  $n$  leg position pairs (positions of left and right legs) detected by a 2D LiDAR in  $T$ . We carefully design a deep neural network (DNN) model as the intention detection function  $h$ .

### B. Gait Dataset Composition

We collect two gait datasets for learning the walking intention detection DNN. The first dataset  $\mathbb{D}$  is collected when a user pushes the walker along. Each data sample  $D_T$  in  $\mathbb{D}$  contains the feet orientation image sequence  $G_T$  and the leg position sequence  $P_T$  within a time window  $T$ , consisting of  $n$  frames.  $n$  is set to a suitable constant value, e.g. 10, as in our experiments with a data sampling frequency of about 4 Hz. The time window  $T$  is about 2.5 seconds which keeps sliding at one-frame intervals. The label, i.e., walking intention, of each sequence sample is set based on the velocity and position change of the walker. Each sequence is the data of the time window  $T$  in which the first and the last frames of the  $T$  are the start and end time points of the sequence.

The other dataset  $\mathbb{S}$  is collected when the user keeps a posture at one spot and the walker is placed static in different positions surrounding the user, to capture different static gait images and leg positions. Each tuple of static gait image and leg positions replicate themselves and form a sequence sample  $S_T$  of length  $n$ . Such replication ensures that the samples in  $\mathbb{S}$  are of the same dimension of samples in  $\mathbb{D}$ . This dataset is intended for improving walking intention detection in the case of specific gait sequences that are under-represented in  $\mathbb{D}$ . For example, data collected when placing the walker on a leftward angle to the user when the user keeps the posture of stepping toward the right emulates the case that the walker mistakenly turns left when the user turns right. Samples in  $\mathbb{S}$  are labeled manually based on the emulated walking intention of the user.

The two datasets are shuffled and combined to form the dataset  $\mathbb{C}$  for learning our walking intention detection DNN model. We also create separate testing datasets with gait data collected from different users, to evaluate our model's generalizability on unseen gait data.

## IV. FFLNET

We design a DNN, the *Front-Following Net (FFLNet)* to estimate the walking intention of the user of the smart walker, for guiding the movement of the walker over time. As mentioned, human walking is a consecutive process; the gait data includes both spatial and temporal features. *FFLNet* combines CNN, and LSTM networks for extracting useful spatial and temporal features. Especially, it contains a *Current Net* and a *Tendency Net*, as illustrated in Fig. 2.

The input to the *FFLNet* is a sequence of gait data  $C_T = \{G_T, P_T\}$  in a time window  $T$ , with  $n$  data points. The input sequence is further divided into two parts, the *Tendency part*

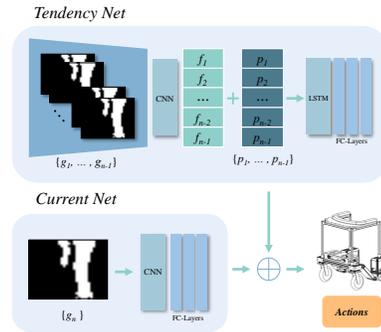


Fig. 2. Overview of FFLNet.

$C_A$  and the *Current part*  $C_B$ .  $C_A$  consists of all data from  $C_T$  except for the latest frame  $(g_n, p_n)$  in the sequence, while  $C_B$  contains the latest frame of gait data only, i.e.,  $C_A = \{(g_1, p_1), (g_2, p_2), \dots, (g_{n-1}, p_{n-1})\}$  and  $C_B = \{(g_n, p_n)\}$ .

The *Tendency Net* takes  $C_A = (G_A, P_A)$  as input, where  $G_A$  is the thermal image sequence  $\{g_1, g_2, \dots, g_{n-1}\}$  and  $P_A$  is the leg position sequence  $\{p_1, p_2, \dots, p_{n-1}\}$ . First, each gait image  $g_i$ ,  $1 \leq i \leq n-1$ , is fed to a CNN for spatial feature extraction; the extracted features are then concatenated with the leg positions  $p_i$  collected at the same time point. Next, the concatenated feature vectors, corresponding to different  $i$ 's, are fed into an LSTM layer for temporal feature encoding. The LSTM layer is further followed by several fully connected layers to produce a feature vector  $f_A$ . The *Tendency Net* architecture design is to conduct a deep computation on each frame in  $C_A$  and to extract their temporal features properly.

The *Current Net* takes  $g_n$  in  $C_B$  as input, which is the last thermal image in the input sequence.<sup>1</sup> The *Current Net* is composed of a CNN followed by several fully connected layers, which outputs another feature vector  $f_B$  of the same length as feature vector  $f_A$ . The CNN structure of the *Current Net* is set the same as the CNN structure of the *Tendency Net*, similarly to extract the spatial feature, together with the following fully connected layers. The rationale of separating the last frame information and use a dedicated *Current Net* to learn features in the last frame is due to our observation that the last frame can often represent the walking intention better than the previous frames.

The two feature vectors  $f_A$  and  $f_B$  are summed up, and fed to a fully-connection layer to produce the final prediction vector  $f_F$ , through a *Softmax* activation function:  $f_F = \text{Softmax}(f_A + f_B)$ .  $f_F$  is a 6-dimensional vector, indicating the probabilities of 6 possible walking intentions: stopping, moving straightly forward, turning left while moving forward, turning right while moving forward, turning left on the spot, and turning right on the spot. The difference between the turning left/right while moving forward and the turning left/right on the spot is that the turning radius of the former is typically larger than one third of the width of the walker, while the turning radius of the latter would be smaller than that (according to our observations). The turning center is

<sup>1</sup>Leg position  $p_n$  in  $C_B$  will be used in the walker control policy.

the midpoint of the connection between the two rear wheels. We do not have moving backward as one output walking intention of *FFLNet*, as it is easy for the walker control module to recognize whether the user is walking backward, based on his/her position relative to the walker detected by the LiDAR. Details of different action modes of the walker, upon user walking intention detection, will be discussed in the next section. We train *FFLNet* using supervised training with the dataset  $\mathbb{C}$  and the Cross-Entropy Loss function [28].

## V. MOVEMENT CONTROL

The specific range for the close proximity in our scenario is that the walker would be in an area within 45cm from the user. That requires the control policy to react to the user's movement promptly while keeping a safe distance from the user. The *FFLNet* offers the timely walking intention prediction. Control policy would rely on the LiDAR data to further confirm the control command to achieve the safe close-proximity following. According to the LiDAR information, we can get the position of the left leg  $l_l(X_l, Y_l)$  and the right leg  $l_r(X_r, Y_r)$ . In addition, the central position of the human  $l_m(X_m, Y_m)$  can be calculated as the midpoint of the connection between the positions of the two legs  $l_l$ ,  $l_r$ , to assist in walker movement control.

We use linear velocity and turning radius to control the movement of the walker. When the linear velocity and turning radius are determined, the angular velocity will also be determined. The linear velocity  $V_{curr}$  is preset according to the user's walking speed, varies from 10 cm/s to 30 cm/s. From experiments, we observe that when the turning radius of the walker is between  $\frac{L}{3}$  and  $2L$  (two times of  $L$ ), the walker can cope with most usage scenarios, e.g., walking through narrow door frames, sharp turning indoors, and walking in long aisles, etc.  $L$  is the width of the walker. In our design, the  $L$  is about 66cm.

Let the LiDAR position be the original  $O$ . The y-axis is along the walker's central axis and the x-axis is perpendicular to the y-axis (Figure 3). Based on the LiDAR measurements, the walking area of the user is divided into the left area ( $X_{min}, X_{cl}$ ), the central area ( $X_{cl}, X_{cr}$ ) and the right area ( $X_{cr}, X_{max}$ ) along the x-axis. In each area, the maximal value along y-axis is the distance from the center axis of the rear wheel to the LiDAR. Here  $X_{max}$  and  $X_{min}$  are the maximum and minimum positions that the user's leg can reach along the x-axis inside the walker. Their distance to the y-axis is 25cm in our design.  $X_{cl}$  and  $X_{cr}$  are the left and right boundaries of the central area along the x-axis.

According to the output of the *FFLNet* and leg position information, the control of the walker is considered in five cases:

- When the *FFLNet* indicates turning right or left on the spot and the central position of human is in the central area of the walker, the rotational speed of the walker is set to a preset constant value  $V_m$  which is 0.2 rad/s.
- When *FFLNet* produces a turning right or left intention and the central position of human is in the right or left area, the turning radius is related to the position of the leg off the

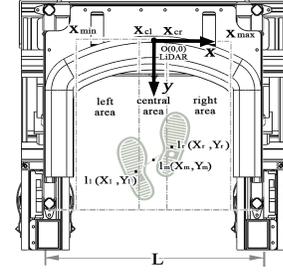


Fig. 3. Division of walking area inside the walker.

central area. The farther the leg deviates from the center area, the smaller the turning radius of the walker should be.  $R_r$  ( $R_l$ ) and  $\omega_r$  ( $\omega_l$ ) are the turning radius and angular velocity when turning right (left), respectively. The turning radius is calculated as follows:

$$R_r = \frac{L}{3} + \frac{5L}{3} \cdot \left(1 - \frac{X_r - X_{cr}}{X_{max} - X_{cr}}\right) \quad (1)$$

$$R_l = \frac{L}{3} + \frac{5L}{3} \cdot \left(1 - \frac{X_{cl} - X_l}{X_{cl} - X_{min}}\right) \quad (2)$$

For the calculation of  $R_r$  in equation 1, since the  $X_r$  would be in the area of ( $X_{cr}, X_{max}$ ), the  $R_r$  would be in the range of ( $\frac{L}{3}, 2L$ ). Same procedure can be adapted to the calculation of  $R_l$ . Then, we can calculate the angular velocities  $\omega_r, \omega_l$  by having  $V_{curr}$  be divided by  $R_r, -R_l$ :

$$\omega_r = \frac{V_{curr}}{R_r} = \frac{3 \cdot V_{curr} \cdot (X_{max} - X_{cr})}{L \cdot (6 \cdot X_{max} - X_{cr} - 5 \cdot X_r)} \quad (3)$$

$$\omega_l = -\frac{V_{curr}}{R_l} = -\frac{3 \cdot V_{curr} \cdot (X_{cl} - X_{min})}{L \cdot (5 \cdot X_l - 6 \cdot X_{min} + X_{cl})} \quad (4)$$

- When the output of *FFLNet* indicates a moving straight forward intention and the central position of human is in the central area, the turning radius is set to  $+\infty$ , the angular velocity is set to 0, and the linear velocity is set to  $V_{curr}$ , to ensure that the walker travels straight ahead at speed  $V_{curr}$ .
- When the output of *FFLNet* indicates stop and the position of human is  $d_s$  (generally about 0~15 cm) away from the walking area along the positive y-axis direction or there is no human detected, linear velocity, angular velocity, and turning radius are all set to 0, to make the walker stop.
- When the position of the user is  $d_b$  (generally about 15~20 cm) away from the walking area along the positive y-axis direction, it shows the user is moving backward. The linear velocity is set to  $-V_{curr}$ , and angular velocity and turning radius are set to 0, to ensure that the walker moves backwards at speed  $-V_{curr}$ .

## VI. EXPERIMENTS AND RESULTS

### A. Implementation Details

1) *Sensors on the Walker*: We use a thermal camera with a  $24 \times 32$  resolution to capture thermal gait images, with a data collection frequency of about 4 Hz. The 2D LiDAR adopted has a sampling frequency of 7.5 Hz. We use electronic wheels and an Inertial Measurement Unit (IMU) to record the movement of the walker.

TABLE I  
DIFFERENT WALKING PATTERNS

Pattern	Description
8-Shape	Walking in an 8-shape path
Forward Turning (Left or Right)	Turning in a circle anti-clockwise or clockwise
Spot Turning (Left or Right)	Turning on the spot towards left or right
Standing Still	Not moving

2) *Gait Data Collection*: We invite different users to collect their gait data for training the *FFLNet*. All the participants are aware and agree that their gait data is used for the research in this project only. The gait data is anonymous and unprofitable so data privacy and safety are guaranteed. To get different gait data, we select users with different gender, height, and weight. The height range is from 160cm to 180cm while the weight range is from 50kg to 75kg. We have 5 users to walk with the walker around for about 20 minutes each, in different patterns as defined in Table I. We ask each user to move in three speed levels in each walking pattern: 0.05 to 0.1m/s, 0.1 to 0.3m/s, and 0.3 to 0.5m/s. We also collect data in the case that no user is using the walker, to allow *FFLNet* to recognize whether a user is inside the walker.

To collect static gait images (set  $\mathcal{S}$ ), we let the users stand in different postures indicating different walking intentions of moving straight forward, turning left/right while moving forward, turning to left/right on the spot, or standing still. In each case, we capture the thermal gait images and the leg position data, and manually label the images. We collect about 400 images from each user.

For testing dataset, we further collect about 15 minutes of gait data from 3 users who do not participate in training data collection. They push the walker following a random path including all kinds of walking patterns as listed above, with different walking speeds.

3) *FFLNet Implementation*: In the *Tendency Net* of *FFLNet*, the CNN contains 4 convolution layers, each with 20 filters of the size of  $3 \times 3$ . The LSTM layer has 64 units, followed by 3 fully-connected layers which consist of 128, 256, and 64 units, respectively. The *Current Net* contains 4 convolution layers and 3 fully-connected layers, using the same layer settings as respectively in the *Tendency Net*. The learning rate is set to  $10^{-5}$ .

We set the length  $n$  of each gait data sequence to 10, as the walking tendency of a user can normally be told by observing the past one to two steps, the time for a user to complete a full step is about 1 to 2 seconds, and the sampling frequency of the thermal camera is about 4Hz. The time window for collecting a sequence of gait data,  $T$ , is set to 2.5 seconds.

### B. Evaluation of *FFLNet*

We compare the walking intention estimation performance of *FFLNet* with two other NNs, which are revised from the two sub-networks of *FFLNet*: (1) *Tendency<sup>+</sup>* is based on the *Tendency Net* in *FFLNet*, by using all  $n$  frames of gait data instead of the first  $n - 1$  frames as the input and adding an output layer to obtain the intention prediction from

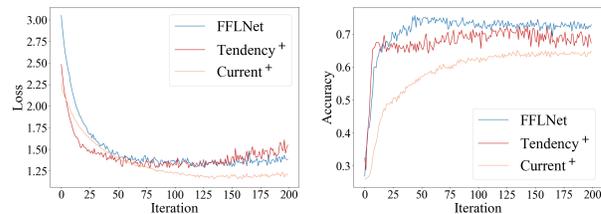


Fig. 4. Validation loss and accuracy during the training of *FFLNet*, *Tendency<sup>+</sup>* and *Current<sup>+</sup>*.

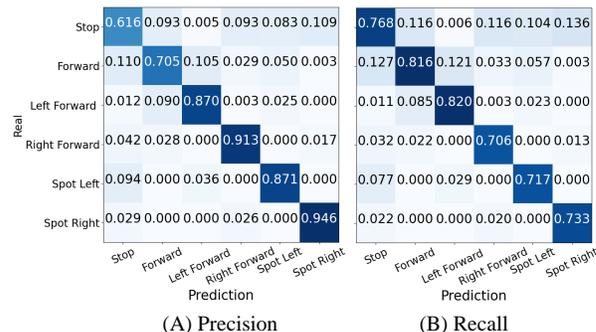


Fig. 5. Confusion matrix of *FFLNet*. (A) Precision matrix; (B) Recall matrix

the features produced by the *Tendency Net*; (2) *Current<sup>+</sup>* is constructed by adding an output layer to the *Current Net* of *FFLNet*. Fig. 4 gives the validation loss and accuracy of the three NN models.

We observe that *Tendency<sup>+</sup>* achieves an accuracy of about 70.7%, while *Current<sup>+</sup>* about 64.5%, which indicates the importance of the temporal features in walking intention detection. *FFLNet*, which combines the *Tendency Net* and the *Current Net*, achieves the best accuracy of 77.4%. This result shows that combining both temporal and spatial features, and emphasizing the last frame of the gait sequence, can lead to better intention detection performance.

We further inspect the intention estimation performance of *FFLNet* by calculating its *Confusion Matrix* [29] which gives the precision and the recall of *FFLNet* in predicting each walking intention class on the test dataset. The results are given in Fig. 5, where “Prediction” is the predicted label and “Real” indicates the true label. Among the six walking intentions, we can see from the precision that *FFLNet* works well in predicting turning left/right while moving forward and turning left/right on the spot. Lower performance results in inferring stopping and moving forward. The reason is that the gait information is similar in these two cases: when moving forward, chances are high for the two legs to be close to each other, which is similar to the posture of standing still, especially when the user is walking at a slow speed with small steps. The error can be reduced by calculating the relative position of the user from the LiDAR data. Further, in the case of distinguishing spot turning and stopping, our model works well with similar leg position data in these situations. While for the recall evaluation, all the intention reaches over 70%, meaning that the *FFLNet* can well distinguish similar intention into the same prediction.

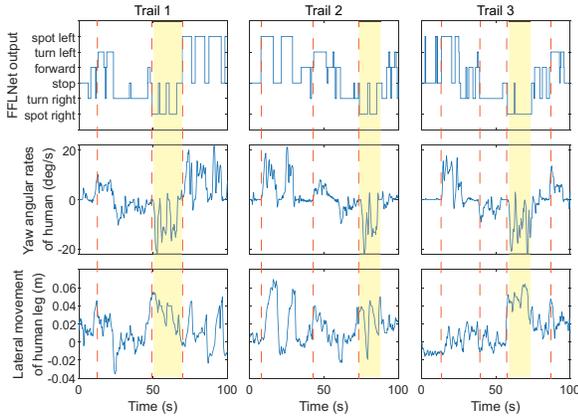


Fig. 6. FFLNet output, yaw angular rates, and lateral movements of human legs.

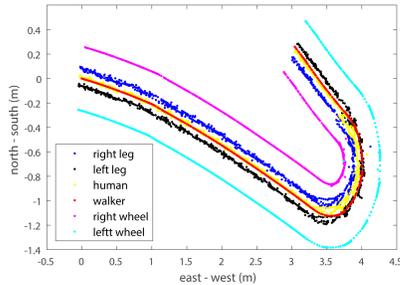


Fig. 7. Trajectory of the smart walker, the human center, the left and right legs, the right and left wheels of the walker.

### C. Online Evaluation

We test online front-following performance by having the walker follow the human automatically along an arbitrary path for about 100s. Three trails of this test are conducted, and the human walks with six walking intentions in arbitrary combination. We evaluate *FFLNet* output against the user’s yaw angular rates and lateral movement of the central position of user legs  $l_m$ . The yaw angular rates are measured by IMU attached to the user waist. The lateral movement of human legs with respect to the walker are measured by the LiDAR data. Fig. 6 shows that *FFLNet* estimations and real turning directions are kept in sync and consistent in orientation. On the other hand, the lateral movements of human legs are sometimes inconsistent with the turning intentions (yellow region in Fig. 6). This shows that the scanning results of the LiDAR are not sufficient for recognizing all walking intentions, and the thermal images are needed. The combination of LiDAR and thermal camera data effectively improves the accuracy of gait recognition.

### D. Evaluation of Movement Control

We let a user walk with the walker without holding the handrails. We apply an IMU on the walker to record its position and direction in real-time (for performance evaluation purpose only). The positions of legs are collected by LiDAR. Then we calculate the coordinates of the legs and wheels to the absolute coordinate system, to evaluate the tracking performance of the walker with the user. Fig. 7 plots the positions obtained for the walker and the user. We observe that the distance between the user and walker trajectories is

much less than  $\frac{l}{2}$ , and the two legs are always within the walking area of the walker, revealing safe and close front-following of the user by the walker is achieved. We further test our walker in an outdoor environment with brick roads and blind tracts. The outdoor temperature will affect the thermal images greatly, bringing a challenge to the *FFLNet*. When the temperature is over 29 Celsius degrees, the *FFLNet* prediction accuracy will drop to about 60% on average. But since the control policy considers the user’s position detection by the LiDAR, the performance of outdoor can still maintain a smooth following with close proximity that the walker is within the range of 40cm away from the user. Though the walker’s shaking in the rough environment can bring errors in the LiDAR measurement, the average error of the leg position is less than 3cm, which can be tolerated.

### E. Comparison with Other Designs

Compared to other designs, our walkers outperform in a close-proximity following while maintaining fast computation time. The computation time for our model to handle a sequence of 10 frames data would be less than 30ms, which is faster than the 500ms for the temporal model of [6]. The turning radius in our walker can be 0cm to achieve higher flexibility, while in [5] the minimum turning radius is 15cm, leading to a worse moving trajectory. For the walker in our previous design [11], the prediction label is labeled as the orientation and movement of the user’s feet. This design could well distinguish the legs’ movement. However, the operation for the old walker is a step-by-step procedure. Also, the old walker cannot predict well when the user just changes their feet orientation without stepping them out. In the new design, we further collect data within steps, enabling the model to predict continuously. The average orientation error between the user and the walker decrease to  $4^\circ$  from the  $5.5^\circ$  in [11].

## VII. CONCLUSION

We design a deep learning model, *FFLNet*, to detect user walking intention for developing a smart walker with automatic user front-following functionality. By exploiting the user’s leg positions and thermal images of the user’s feet, the proposed approach can successfully detect more types of walking intentions than other existing smart walker designs do. Our deep learning model extracts spatial and temporal features of gait sequences, and achieves good detection accuracy among representative walking intentions; combined with the walker control model, good front-following tracking results are attained. However, the proposed method has some limitations. One limitation is that the thermal camera we use is greatly affected by the environmental temperature. In a warm place, it’s hard to distinguish the user’s legs from the background because their temperature is similar. Under this condition, the walker has to rely on the LiDAR data only to detect human position for front-following. Another limitation of our proposed approach is that our learning dataset is small and not diverse enough. More user gait data need to be collected to get a more universal model in the future.

## REFERENCES

- [1] W. H. Organization *et al.*, *World health statistics 2021: monitoring health for the SDGs, sustainable development goals*. World Health Organization, 2021.
- [2] S. Karimi, A. Soroush, F. Towhidi, B. R. Makhsofi, M. Karimi, S. Jamehshorani, A. Akhgar, M. Fakhri, and A. Abdi, "Surveying the effects of an exercise program on the sleep quality of elderly males," *Clinical interventions in aging*, vol. 11, p. 997, 2016.
- [3] M. M. Martins, A. Frizzera-Neto, E. Urendes, C. dos Santos, R. Ceres, and T. Bastos-Filho, "A novel human-machine interface for guiding: The neosmas smart walker," in *2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*. IEEE, 2012, pp. 1–7.
- [4] W.-C. Cheng and Y.-Z. Wu, "A user's intention detection method for smart walker," in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2017, pp. 35–39.
- [5] S. D. Sierra M, M. Garzón, M. Múnera, C. A. Cifuentes *et al.*, "Human-robot-environment interaction interface for smart walker assisted gait: Agora walker," *Sensors*, vol. 19, no. 13, p. 2897, 2019.
- [6] M. Palermo, S. Moccia, L. Migliorelli, E. Frontoni, and C. P. Santos, "Real-time human pose estimation on a smart walker using convolutional neural networks," *Expert Systems with Applications*, vol. 184, p. 115498, 2021.
- [7] M. F. Jiménez, M. Monllor, A. Frizzera, T. Bastos, F. Roberti, and R. Carelli, "Admittance controller with spatial modulation for assisted locomotion using a smart walker," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 3, pp. 621–637, 2019.
- [8] G. Lee, T. Ohnuma, and N. Y. Chong, "Design and control of jaist active robotic walker," *Intelligent Service Robotics*, vol. 3, no. 3, pp. 125–135, 2010.
- [9] W.-H. Mou, M.-F. Chang, C.-K. Liao, Y.-H. Hsu, S.-H. Tseng, and L.-C. Fu, "Context-aware assisted interactive robotic walker for parkinson's disease patients," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 329–334.
- [10] C. Valadão, E. Caldeira, T. Bastos-Filho, A. Frizzera-Neto, and R. Carelli, "A new controller for a smart walker based on human-robot formation," *Sensors*, vol. 16, no. 7, p. 1116, 2016.
- [11] X. Zhao, Z. Zhu, M. Liu, C. Zhao, Y. Zhao, J. Pan, Z. Wang, and C. Wu, "A smart robotic walker with intelligent close-proximity interaction capabilities for elderly mobility safety," *Frontiers in Neurobotics*, vol. 14, p. 75, 2020.
- [12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [13] W. Chung, H. Kim, Y. Yoo, C.-B. Moon, and J. Park, "The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder," *IEEE transactions on industrial electronics*, vol. 59, no. 8, pp. 3156–3166, 2011.
- [14] Q. Xiao, F. Sun, R. Ge, K. Chen, and B. Wang, "Human tracking and following of mobile robot with a laser scanner," in *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2017, pp. 675–680.
- [15] F. Hoshino and K. Morioka, "Human following robot based on control of particle distribution with integrated range sensors," in *2011 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2011, pp. 212–217.
- [16] J. Yuan, S. Zhang, Q. Sun, G. Liu, and J. Cai, "Laser-based intersection-aware human following with a mobile robot in indoor environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [17] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A novel vision-based tracking algorithm for a human-following mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1415–1427, 2016.
- [18] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Person following robot using selected online ada-boosting with stereo camera," in *2017 14th conference on computer and robot vision (CRV)*. IEEE, 2017, pp. 48–55.
- [19] R. Algabri and M.-T. Choi, "Deep-learning-based indoor human following of mobile robot using color feature," *Sensors*, vol. 20, no. 9, p. 2699, 2020.
- [20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [21] I. Condés, J.-M. Cañas, and E. Perdices, "Embedded deep learning solution for person identification and following with a robot," in *Workshop of Physical Agents*. Springer, 2020, pp. 291–304.
- [22] I. Condés and J. M. Cañas, "Person following robot behavior using deep learning," in *Workshop of Physical Agents*. Springer, 2018, pp. 147–161.
- [23] X. Wang and W. Q. Yan, "Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory," *International journal of neural systems*, vol. 30, no. 01, p. 1950027, 2020.
- [24] H. Arshad, M. A. Khan, M. I. Sharif, M. Yasmin, J. M. R. Tavares, Y.-D. Zhang, and S. C. Satapathy, "A multilevel paradigm for deep convolutional neural network features selection with an application to human gait recognition," *Expert Systems*, p. e12541, 2020.
- [25] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 2, pp. 316–322, 2005.
- [26] T. Steinmetzer, I. Bonninger, B. Priwitzer, F. Reinhardt, M. C. Reckhardt, D. Erk, and C. M. Travieso, "Clustering of human gait with parkinson's disease by using dynamic time warping," in *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOB)*. IEEE, 2018, pp. 1–6.
- [27] V. G. Torvi, A. Bhattacharya, and S. Chakraborty, "Deep domain adaptation to predict freezing of gait in patients with parkinson's disease," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1001–1006.
- [28] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.
- [29] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection." *MAICS*, vol. 710, pp. 120–127, 2011.