sensors

MDPI

Article

EL-SLE: Efficient Learning Based Stride-Length Estimation Using a Smartphone

Mingcong Shu, Guoliang Chen * and Zhenghua Zhang

School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 21116, China
* Correspondence: chgl_cumt@163.com

Abstract: The pedestrian stride-length estimation is a crucial piece of personal behavior data for many smartphone applications, such as health monitoring and indoor location. The performance of the present stride-length algorithms is suitable for simple gaits and single scenes, but when applied to sophisticated gaits or heterogeneous devices, their inaccuracy varies dramatically. This paper proposes an efficient learning-based stride-length estimation model using a smartphone to obtain the correct stride length. The model uses adaptive learning to extract different elements for changing and recognition tasks, including Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) modules. The direct fusion method maps the eigenvectors to the appropriate stride length after combining the features from the learning modules. We presented an online learning module to update the model to increase the SLE model's generalization. Extensive experiments are conducted with heterogeneous devices or users, various gaits, and switched scenarios. The results confirm that the proposed method outperforms other state-of-the-art methods and achieves an average 4.26% estimation error rate in various environments.

Keywords: indoor positioning; stride-length estimation; CNN; LSTM; adaptive learning; smartphone sensors

check for updates

Citation: Shu, M.; Chen, G.; Zhang, Z. EL-SLE: Efficient Learning Based Stride-Length Estimation Using a Smartphone. *Sensors* **2022**, *22*, 6864. https://doi.org/10.3390/s22186864

Academic Editors: Ying-Ren Chien, Mu Zhou, Liang-Hung Wang and Xun Zhang

Received: 24 June 2022
Accepted: 6 September 2022
Published: 10 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Reliable stride-length estimation is essential for many applications, including health monitoring and Internet of Things (IoT) services [1–3]. Moreover, accurate Stride-Length Estimation (SLE) plays a necessary procedure in the Pedestrian Dead Reckoning (PDR) mechanism in indoor positioning research [4,5]. Significantly, PDR using smartphones' Micro-Electro-Mechanical System (MEMS) is more flexible than the dedicated device, such as shoes' and shank's IMU. Regarding the PDR mechanism with a smartphone, stride length calculates the location directly, so the results determine the accuracy of the positioning system [6]. However, dynamic environments, motion states, and posture significantly express different inertial information. The low-cost MEMS are inherent with noticeable random noise, making it challenging to obtain robust stride-length estimation with off-the-shelf smartphones.

Many studies about pedestrian stride-length estimation are available in the literature. They show good performance with some assumptions, such as single scenario and average speed, resulting in the low generalization of the model in real life. Some studies have proposed an SLE model based on the users' leg length, gender, or weight information. However, those methods manually set personalized parameters from the empirical formula [7–9].

To address the issue of changing states and devices or user heterogeneity, we proposed efficient learning-based stride-length estimation (EL-SLE) by using adaptive learning and online learning. The deep neural network can autonomously map the measurements to predicated results, and the task-orientation deep neural network (DNN) design is vital to better mapping functions. Inspired by the network for camera pose estimation [10], we

proposed adaptive learning to extract multiple features for recognition tasks and temporal changing awareness tasks by using the CNN network and LSTM network, respectively. Furthermore, the online learning module updates the parameters of the CNN framework to improve the generalization of the SLE model in the new scenes. To the best of our knowledge, we are the first to propose an SLE model based on both CNN and LSTM to extract multiple features with an adaptive learning framework. Moreover, visual-based localization is adopted to obtain the ground truth for training data labeling without additional hardware. Our key contributions are as follows:

- We cast the stride-length estimation problem as an adaptive learning problem with multiple feature representations using CNN and LSTM modules. We leverage data preprocessing on the IMU measurements with autoencoders and preintegration to eliminate white noise and improve the efficiency of the network.
- We propose a convenient training data obtaining method based on visual-inertial odometry, determining accurate labels of IMU training data for the SLE model using a smartphone's built-in camera.
- We proposed an online learning module that recognizes the current motion characters to update the network, automatically adjusting the mapping function for the user's stride-length model using the aid of visual-inertial odometry.
- We conduct extensive experiments with various scenarios, motion modes, gaits, and devices/users, and the results show that the EL-SLE model outperforms the state-of-the-art methods.

The paper is organized as follows: Section 2 presents an overview of related work; Section 3 describes the proposed efficient learning-based stride-length estimation (EL-SLE) model by using adaptive learning and an online learning module; Section 4 verifies the performance of the proposed model with elaborate experiments; and Section 5 presents the conclusions and future work discussion.

2. Related Work

Accurate stride-length estimation is essential for field applications such as human motion monitoring, gait analysis for the elderly, and IoT service [11,12]. At the same time, various solutions, such as Vicon using the camera, depth-sensing camera, pressure sensor, etc., can provide the stride length [13–16]. However, the inertial sensor seems more suitable for SLE in our daily lives due to its low implementation cost and unlimited range of motion. This section briefly overviews some related works about inertial sensor-based stride-length estimation.

Inertial sensors are ubiquitous in mobile devices, and they can derive kinstate based on Newton's second low or empirical formula. The SLE using inertial sensors is usually divided into two different approaches [17]: direct approaches based on double integration and indirect approaches based on a symmetrical gait model. The double acceleration integration is used to derive the step length using the traditional motion mechanism, and the one is to find how to establish the relationship between the statistical data with gait models and stride length. Regarding model theory and convenience, the direct approach may have more advantages than the indirect approach due to requiring no training data. However, noise and bias are inherent with the low-cost inertial sensors, resulting in cumulated error during the integration process. Therefore, the motion character of pedestrian walking is analyzed using heuristics to constrain the cyclical motion. Zero Velocity Update (ZUPT) has been proposed to reduce the error accumulation by resetting the motion vector when the foot is touching the ground during pedestrian walking [18–20]. ZUPT is employed with the foot-mounted sensor for better application, and the effect for the other body parts (such as pocket or handheld) is not significant during complex movement. Therefore, the direct-based stride-length estimation in the mobile device is not reliable in obtaining an accurate calculation using the integration methods.

According to the motion statistics and assumption for the stride-length estimation, the indirect approaches can be classified as empirical relationships [7,8], inverted pendu-

lum [21,22], and statistical regression method, including acceleration-based methods [23,24], step frequency-based methods [25–27], angle-based methods [28,29] and multiparameter methods [30,31]. The training data collection and matching process are needed in these methods to learn the relationship between sensor information and the stride length. Moreover, the cyclic motion features are analyzed to obtain an accurate step length model [32] to remove the limitation of carrying the device for pedestrians. However, the extracted features tend to be different, and they cannot provide the generalization performance with the different pedestrians' motions. Similarly, a context-based stride-length estimation is proposed to obtain motion features using a linear fusion of the stride frequency and acceleration variance [33]. The fatal stride-length error may come with the error of context recognition, and a context weighted model is proposed to estimate the stride length, and the different context probabilities are calculated as weight to compute the length [30]. Nonetheless, the manual classification based on prior information cannot cover all the pedestrian motion statement and scenarios.

Recently, regression-based and deep learning-based methods have been proposed to obtain accurate stride-length estimation [28,34–36]. The smartphone carrying modes are recognized to increase the robustness of the calculation [34]. For more training data and a reliable model, GPS information is used to label the data from the inertial sensors during pedestrian walking, and a hybrid multiscale CNN and recurrent neural network (RNN) are employed to regress the speed and time interval [35]. However, the model cannot be applied in indoor scenes due to lacking great GPS information for indoor data. According to sequential features, the denoising autoencoders and LSTM predicate the stride length [37]. Another solution for stride-length estimation is proposed by only using CNN [36]. However, these methods depend on the training data with pre-set walking characteristics which cannot apply to all gaits and environments.

Moreover, none have paid spatial and temporal features corresponding to recognition and modeling problems. The purpose of this paper is to employ a deep neural network to extract multiple features for the stride-length estimation, and the online learning module increases the robustness of the model by considering the heterogeneity of devices and users. Meanwhile, training data is required in the learning-based methods, and two methods are usually employed: segmentation with known total distance [9] and additional device aiding [37]. The first method is simple with accuracy, while it is not suitable for complex training in a large-scale environment. The other method requires an additional device, such as an optical motion capture system [38], a specialized Optojump system [39], and reliable foot-mounted sensors [40]. However, these methods require specific infrastructure and expensive devices. In this paper, the smartphone's built-in camera obtains the ground truth using the visual-inertial odometry (VIO) solution [41,42], which is convenient for labeling the training data requirements.

We propose an efficient learning-based SLE model that combines adaptive learning and an online learning module to obtain a robust stride-length estimation with a smartphone. As shown in Figure 1, we take advantage of the neural networks to extract the statistic features and temporal features for multiple representations. Moreover, online learning maintains the model's generalization for heterogeneous devices or users. In addition, we proposed a convenient method to obtain the training data for the SLE model using smartphones.

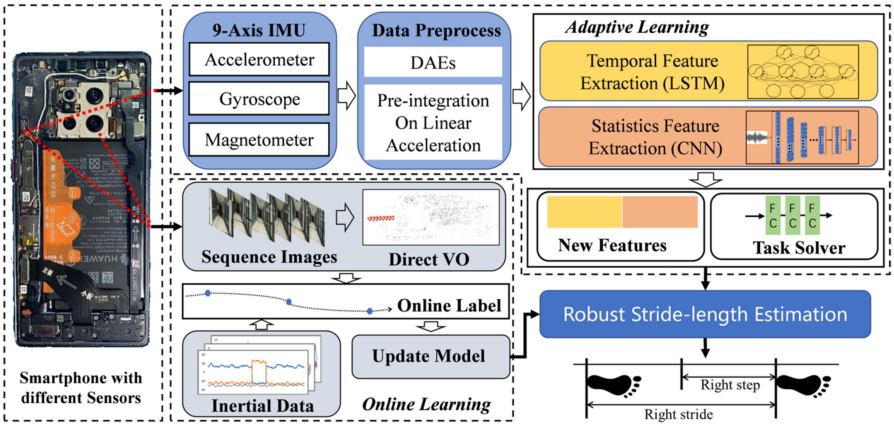


Figure 1. Flowchart of the efficient learning-based stride-length estimation.

3. Algorithm Description

3.1. System Architecture

In this paper, we consider a robust stride-length estimation scenario where the pedestrian is carrying a smartphone, and their motion has no limitation in complex environments. The learning-based model maps the IMU measurements to the stride length for robust SLE, and the system architecture is shown in Figure 2. Firstly, we leverage denoising autoencoders to achieve data augmentation and employ preintegration to reduce computational complexity. Moreover, the adaptive learning module extracts the statistical and temporal features using CNN and LSTM, respectively. We employ a visual-based localization algorithm for the training data to provide reliable ground truth using only smartphones. Subsequently, we present the online learning module to update the model for improving the generalization of the SLE model in new scenarios.

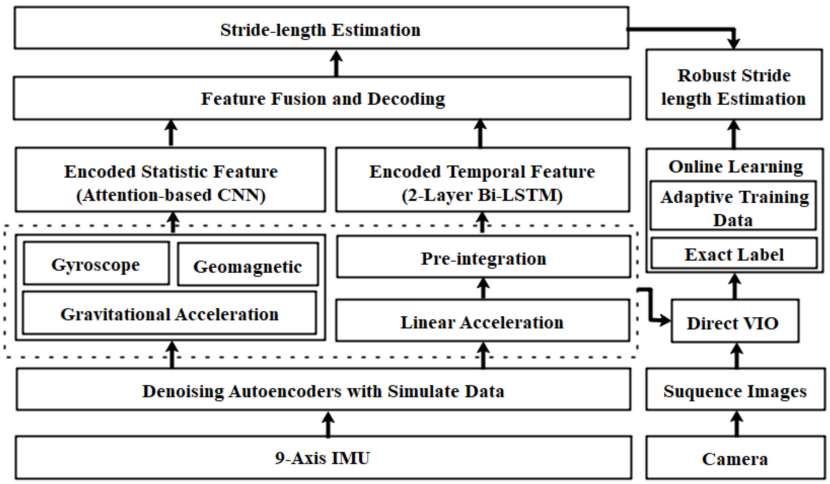


Figure 2. The system architecture of the efficient learning-based stride-length estimation model.

3.2. Stride Length Estimation Based on Adaptive Learning

This section presents the theory and details of the adaptive learning framework for the stride-length estimation model. Our proposed learning framework consists of four parts, which are as follows: (1) data preprocessing that employs data augmentation and data preintegration to improve the efficiency of the network; (2) modified CNN is used to extract the statistical features representation for various scenarios; (3) a bidirectional LSTM network obtains features representation with temporal correlations and continuity constraints; and (4) the features fusion strategy, including a regression model and fully-connected layers, are presented to obtain the outputs of the model.

3.2.1. Data Preprocessing with Denoising Autoencoders and Preintegration

To denoise the sensor readings from the low-cost IMU, the data-driven approach based on an autoencoder denoises motion signals, and we apply data augmentation for a better model with robust features. Furthermore, data preintegration is employed to reduce the time consumption of the feature extraction stage. As shown in Figure 3, the data preprocessing that we proposed consists of three steps: data augmentation, autoencoder, and preintegration. The specific process is shown in the following.

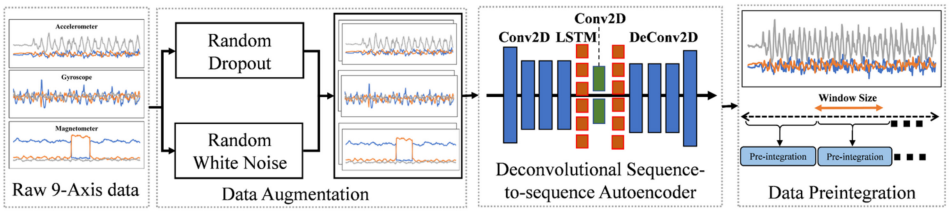


Figure 3. The framework of the data preprocessing.

Considering the time series data from low-cost IMU, we use random dropout and random white noise to obtain the data augmentation [39,43]. The first method is that the part signal information is randomly removed and filled with zero in this part for the autoencoder module. The second method is adding random noise to the dropout part to extract the more significant features from the raw inertial data. The data augmentation can increase the model's noise tolerance for signal reconstruction. Specifically, the accelerometer data and gyroscope data are added to the zero mean random normal noise, while we conduct the random dropout methods for the magnetometer data due to their noise being significantly related to environments [44]. One percent of the IMU data's maximum value determines the noise's standard deviation. Values are set as 0.16 and 20 for the accelerometer and gyroscope, respectively, as shown in the following:

$$\hat{y}_i^x = \begin{cases} y_i^x + N(u = 0, \sigma = 0.16), \; if \; x = Acc \\ y_i^x + N(u = 0, \sigma = 20), \; if \; x = Gyro \end{cases} \tag{1}$$

where y_i^x, $\hat{y}_i^x$ denote the input signal and output x by adding zero mean random noise.

The higher-level representations are extracted to obtain the target results and hidden underlying data-generation relationships to denoise the inertial data efficiently. Inspired by the related work [45] about motion signal reconstruction, we employ the deconvolutional sequence-to-sequence autoencoder to represent the IMU model accurately, improving the precision of pedestrian gesture recognition. As we know, specific information represents the IMU data, and it is challenging to leverage the handcrafted features to show the signal. Therefore, the data-driven approach learns the IMU data. As shown in the figure, in the variational autoencoder framework, we use 200 $\times$ 3 size samples from the IMU data with two kinds of augmentation forms as the input of the autoencoder, including an accelerometer, gyroscope, and magnetometer. For the structure of the neural network, we employ four Conv2D and four De-Conv2D layers with ReLu activation and 32 $\times$ 5 kernels. We use four stride sizes on the first and the last layers. The two stateless 64-unit LSTM layers are applied in the encoder and the decoder and dropout layers with a p-value of 0.5. Therefore, a stack of convolutional neural-network layers (Conv2D) extracts features, and a sequence of LSTM units perceives the temporal relationships. The autoencoder model takes the augmentation data h as input. Then, the encoder $Enc(.)$ maps the data to representational features in the hidden layers, and the decoder $Dec(.)$ maps the lower dimension space to generation data $\hat{h}$. The insight of an autoencoder is how to establish the

optimal encode network that captures the samples' dispersion characteristics and generates the data with the minimum error between input and output.

$$\begin{cases} h = \begin{bmatrix} H(y_i^x) \\ H(\hat{y}_i^x) \end{bmatrix}, x = Acc/Gyro/Mag \\ \hat{h} = Dec(Enc(h)) \end{cases} \tag{2}$$

where h represents the hidden feature from IMU measurements. These higher-level features represent the IMU data due to its noise, and we employ the probabilistic generative model [45] to establish the relationship between features and the time-domain IMU samples in the following:

$$\begin{cases} p_\theta(x,z) = p_\theta(x|z)p(z) \\ p(z) = N(0,\mathrm{I}) \\ p_\theta(x|z) = N(\mu_\theta(z), \sigma_\theta^2(z)\mathrm{I}) \end{cases} \tag{3}$$

where likelihood $p_\theta(x,z)$ quantifies the relationship between the observed IMU samples x and the hidden random variable z, and the prior $p(z)$ quantifies the information about z before seeing samples. Mean $\mu_\theta(z)$ and variance $\sigma_\theta^2(z)$ represent the latent parameters in a neural network. This representation model uses the posterior $p_\theta(x|z)$ to infer z and obtain parameters that maximize the marginalized likelihood $p_\theta(x|z)$. As the encoder model in [45], the theory of variance inference approximates the posterior $q_\phi(z|x)$ with a similar and tractable distribution:

$$q_\phi(x|z) = N(\mu_\phi(z), \sigma_\phi^2(z)\mathrm{I}) \tag{4}$$

To train the parameters of the autoencoder model, we employ a weighted loss function that combines Kullback–Leibler divergence (KL) and Means Square Error (MSE) to train the model, and the representation is as follows:

$$L = r \cdot (E_{MSE}[(\hat{h} - h)^2]) + (1 - r) \cdot E_{KL}[q(z|x)||p(z)] \tag{5}$$

where E_{KL} represents the KL divergence describing the distribution similarity; E_{MSE} measures the error of the input and output signal; and r is the weight values.

After denoising the autoencoder, the preintegration is adopted to improve the efficiency of the neural network, obtaining motion constraint variables by using IMU measurements in a pose graph. The 9D vectors constrain the orientations, velocities, and positions of keyframes based on the mathematical model of the IMU, and the measurements are used to propagate the object's motion in the inertial frame using the recursive physic model in the following:

$$\begin{cases} R_{n+1} = R_n Exp((\widetilde{\omega}_n - b_g - \eta_g)\hat{}\Delta t) \\ V_{n+1} = V_n + g\Delta t + R_n(\widetilde{a} -_n b_a - \eta_a)\Delta t \end{cases} \tag{6}$$

where R_n and V_n are orientation and velocity of the smartphone sensor in the world coordinate system, respectively, and Δt denotes the sampling time of the IMU. $\widetilde{\omega}_n$ and $\widetilde{a}_n$ represent the angular velocity and acceleration measurements from the gyroscope and accelerometer, which are with additive Gaussian noise η_g, η_a and random walk bias terms b_g, b_a, respectively. Furthermore, the $Exp(\cdot)$ function in the above equation is the SO3 exponential map that converts the skew-symmetric members of the lie algebra so3 to their corresponding SO3 matrix, and $(\cdot)\hat{}$ the operator converts a 3D vector into its skew-symmetric matrix representation.

The highly changing motion of the carrier may violate the constant world acceleration from the low-cost inertial sensor. Therefore, the consecutive IMU samples can be assumed to be constant in the body frame. Known as preintegration IMU factors, the consecutive IMU samples are compressed into a single vector using the constraint. Then, the state

transition based on a preintegration constraint from the IMU measurements between time i and j can present as:

$$\begin{cases} \Delta R_{ij} = \prod_{k=1}^{j-1} Exp((\widetilde{\omega}_k - b_k^g - \eta_k^g)\Delta t) \\ \Delta V_{ij} = \sum_{k=1}^{j-1} \Delta R_{ik}(\widetilde{a}_k - b_k^a - \eta_k^a)\Delta t \end{cases} \tag{7}$$

where the initial state terms are moved to the left-hand side of the equation. In this paper, we only extract the pre-integrated features from the acceleration, which is the input in the LSTMs network, and the preintegration for the series signal can reduce the time consumption of the complex neural network without reducing signal quality.

3.2.2. Recognition Feature Extraction with CNN Network

CNN is suitable for human activity recognition using hidden features from the inertial data in traditional works [45,46]. In this paper, the nine-axis data are used as input in the CNN to obtain the recognition features, and the inertial stride curve x with time interval T presents as:

$$\begin{aligned} &x = (x_1, x_2, \cdots, x_t), \\ &x_t = (Acc_t^x, Acc_t^y, Acc_t^z, Gyro_t^x, Gyro_t^y, Gyro_t^z, Mag_t^x, Mag_t^y, Mag_t^z)^T \end{aligned} \tag{8}$$

where Acc, $Gyro$, Mag denote the output from preprocessing, including gravitational acceleration, gyroscope, and magnetometer.

Considering the noise of the low-cost IMU, we employ the modified CNN model [47] to amplify the prominent activity data and alleviate the impact of sensor noise. As shown in Figure 4, the modified CNN framework consists of two auxiliary submodules and an entire CNN pipeline which includes convolutional layers, pooling layers, and fully connected layers. The core of the submodule is the compatibility calculation between the local feature vector extracted in the middle layer of the CNN structure and the global feature vectors. The auxiliary submodule can provide the compatibility calculation for the feature extraction in the CNN pipeline, and the module's detailed effect is how to integrate two vectors from the different features by using a compatibility function with a dot product. Then, a set of feature vectors $L^s = \{l_1^s, l_2^s, \cdots, l_n^s\}$ is from a convolutional layer $s \in \{1, 2, \cdots, n\}$, and a global feature vector G connects with a set of feature vectors by using an additional operation. Then, a dot product is employed to represent the relationship between feature vectors and a weight vector u:

$$c_i^s = \langle u, l_i^s + G \rangle, i \in \{1, 2, \cdots, n\} \tag{9}$$

where c_i^s represents the compatibility score. Then, the normalized form can be obtained by a Softmax function:

$$\begin{cases} a_i^s = \frac{\exp(c_i^i)}{\sum_j^n \exp(c_i^s)} \\ A^s = \{a_1^s, a_2^s, \cdots, a_n^s\} \end{cases} \tag{10}$$

Then, the normalized compatibility score A^s provides a single vector g^s for each layer s by using the element-wise average method:

$$g^s = \sum_{i=1}^{n} a_i^s \cdot l_i^s \tag{11}$$

Subsequently, the calculation value g^s replaces the global feature g, and the new feature vector presents as:

$$g = [g_1, g_2, \cdots, g_n] \tag{12}$$

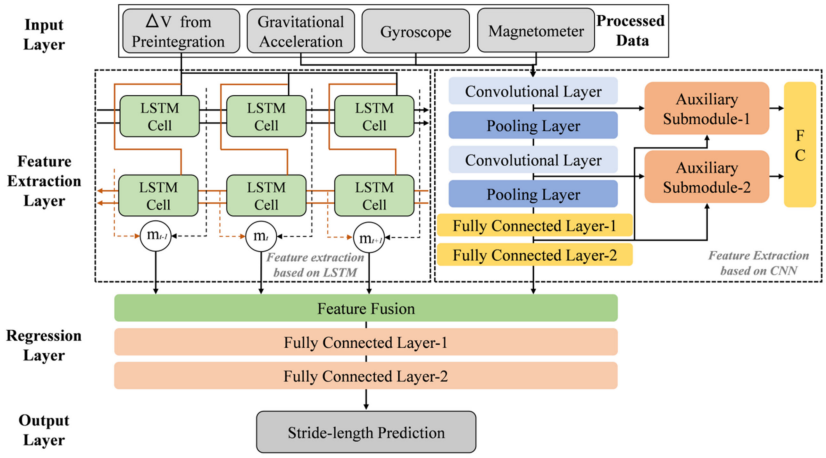


Figure 4. The efficient learning-based model architecture using spatial and temporal features representation for stride-length estimation.

In the auxiliary submodule of the CNN, the compatibility score A^s represents the probability of the region where the pedestrian motion shows significant changes, and the auxiliary submodule represents the feature map. The weighting parameters in the module can enhance salient features and weaken insignificant features. CNN plays a significant role in extracting the statistical features for recognizing different gaits and scenes, and the experimental result in Section 4 has verified the performance.

3.2.3. Temporal Features Extraction with LSTM Network

For the IMU data's temporal features, the LSTM network is designed to handle time-series signals and capture the long-range dependencies in the sequential data. Unlike the direct LSTM network [40], we employ the bidirectional LSTM network to establish the relationship between the current statement and the front or back part statement.

As shown in Figure 4, the bidirectional LSTM network contains two layers, a forward layer and a backward layer, which are composed of the primary LSTM cell [48], and each cell process a sample by the forget gate, input gate, and output gate structure which are identical with the structure of works [49,50]. The input vector and long-term state represent as x_t and c_t at time-step t. h_t and $h_t{}'$ represent the recurrent hidden states of the forward and backward layers on the data sequence. For the hidden features provided by bidirectional layers, we employ the hidden states of the two layers to connect with concatenating setting m_t at the time t, which is taken as the final recurrent hidden state of the network at a time t, as shown in the following:

$$m_t = [h_t, h_t{}'] \tag{13}$$

Then, all recurrent hidden states are combined with the output of the LSTMs layers, which preserve the temporal features at all time steps, and the output matrix O_{LSTM} can be written as:

$$O_{LSTM} = [m_1, m_2, \cdots, m_n]^T \tag{14}$$

where n is the sequence length, and the max-pooling is employed to reduce redundant items for significant features.

3.2.4. Regression Based on the Fusion Features

We now combine the two high-level feature representations by the CNN and LSTM network from the raw IMU data. The extracted temporal and spatial features generate more comprehensive and distinctive fusion features. The direct fusion approach is simpler and more efficient than the soft and hard fusion approaches [51]. Due to the features from the same sensor modality channels, we employ a straightforward approach to obtain the features fusion strategy using multi-layer perceptions (MLPs). The e regression layer

provides the predicated stride length. As shown in Figure 4, once the hidden feature is determined, the nonlinear function has established a map from the feature vectors to the stride length using the training data. The error loss function $L(D, G)$ is how to achieve the minimization for the regression layer between prediction and ground-truth of stride length, as expressed as:

$$\begin{cases} \hat{y} = G(g_{direct}(a_{LSTM}, a_{CNN})) \\ L(D, G) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \end{cases} \tag{15}$$

where $[a_{LSTM}, a_{CNN}]$ denote an MLP function that concatenates features from the adaptive learning module, which are used to represent the sequential and statistical features from the CNN and LSTM network, respectively. y_i denotes the ground truth of the stride length from the input D, $\hat{y}$ denotes the estimation result from the regression layer, and $G(g_{direct}(a_{LSTM}, a_{CNN}))$ represents the regression module for output determination. The loss function we use is standard in training the neural network.

3.3. Vision-Aided Training Data Collection

In this section, we present the data collection process for offline training and confirm the approach's feasibility. There are many ways to obtain the ground truth for labeling data, such as manual segmentation, Vicon System, and dedicated track [38,43,52]. However, these approaches have complicated operations and need high-cost equipment support.

Recently, it has been possible to employ Visual-Inertial Odometry (VIO) with off-the-shelf smartphones [41,42], and this method can provide accurate locations once the loop detection is working. Inspired by the work [53], we propose a convenient approach to collect the training data using arbitrary smartphones. The framework of the collection approach is shown in Figure 5. We fix the smartphone to the chest with the camera facing out for motion tracking, and the other smartphone can be carried naturally, such as handheld gestures, being placed inside a pocket, or carried inside a bag, while the pedestrian conducts movement by normal walking, fast walking, running, or standing in various environments.

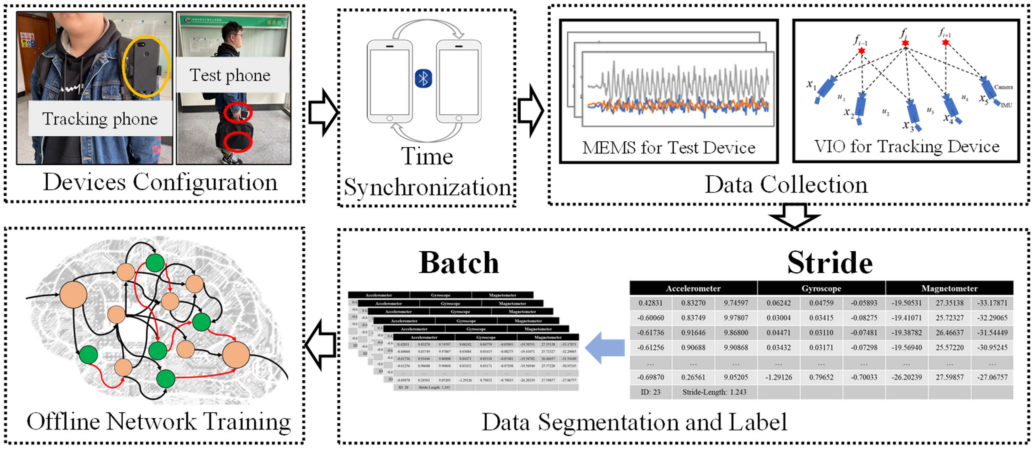


Figure 5. The framework of the vision−aided training data collection.

The proposed approach for training data collection consists of four parts, namely, device configuration, time synchronization, data collection, and data labeling. In the first step, the two smartphones are carefully calibrated, and we operate the process in each data sequence to guarantee the data quality. We correct the bias and scale errors using the method in [54]. Then, the tracking device is fixed to the chest with accessible equipment, as shown in the figure, and the test device is in the pre-set gestures. Secondly, an Android APP we developed is installed on the two smartphones for data collection, and the tracking device can connect with the test device through Bluetooth. Therefore, the system clocks of the device are with time synchronization. Thirdly, the two smartphones conduct the

data collection. The IMU measurements are from the test devices, and the VIO algorithm provides an accurate position from the tracking device. We employ the sampling frequency of inertial sensors, 100 Hz, like the two smartphone settings in [40], and the 50 Hz for the camera frame rate. To generate the segmentation data for training, we employ a peak-based step detection algorithm [55] to count the steps during the motion and to determine the timestamp corresponding to the nodes of VIO. The steps are detected on the time t by using the algorithm. Then the nodes p_k^i of the VIO can be obtained for the ground-truth L_g^K, as expressed in the following:

$$\begin{cases} N = \left\{ (t_i, p_1^i), (t_j, p_2^j), \cdots, (t_n, p_N^n) \right\} \\ p_k^i = (x_K^i, y_K^i, z_K^i, q_K^i) \\ L_g^K = \sqrt[2]{(x_{K+1} - x_K)^2 + (y_{K+1} - y_K)^2} \end{cases} \tag{16}$$

where p_1^i denotes the 6-DoF pose of node detected at K step of time t_i in step detection algorithm.

Finally, we split the IMU data according to the timestamp of the above nodes. The training data is provided by the inertial data and stride length. To generate the fixed size and equal scale of input for the network, we set the 200 samples as the input size to cover various statements of pedestrian walking. For sequences longer than 200, we use the down-sampling strategy to process the original data and obtain the exact size of sequence samples. The ratio of downsampling is not an integer, while it cannot affect the learning process due to the neural network method with high-frequency samples input. They need to be filled with 0 to the samples when the sequences are shorter than 200. In each segmentation for training, the collection data E_K contain 200 samples, nine channels for inertial sensors, and the corresponding ground-truth of stride length, as expressed in the following:

$$\begin{cases} E_K = \left\{ ID_K, (e_1, e_2, \cdots, e_i, \cdots, e_{200}), L_g^K \right\} \\ e_i = \left\{ (Acc_x, Acc_y, Acc_z), (Gyro_x, Gyro_y, Gyro_z), (Mag_x, Mag_y, Mag_z) \right\} \end{cases} \tag{17}$$

where ID_K denotes the identifier of K^{th} the segment, and e_i represents a 9-dimensional vector from the inertial sensor.

After obtaining reliable locations from the convenient VIO-aided collection method, the labeled data are used as the input for network training processing. Figure 6 shows the stride-length estimation results of this approach when the pedestrian walks in the set path with or without loop detection, and the stride length is the same according to the floor tiles. From Figure 6, we find that the VIO can provide accurate length estimation once the loop is detected [42], and this approach needs walking in a loop for better ground-truth obtaining when we collect the training data. As shown in Figure 6, the errors of length estimation are almost less than 5 cm from the VIO with loop detection, which is enough to provide the ground truth for the model training.

3.4. Online Learning Module for Model Updating

The online learning module is presented in this section to update the stride-length estimation model. Online learning can evolve the model to improve the robustness of neural networks [40,56]. It is essential to verify the offline SLE model and update the parameters of DNN for new data. We employ visual localization to obtain the new labeled data, and the model updating framework is performed to analyze the validity of the offline model. The framework of the online learning module is shown in Figure 7.

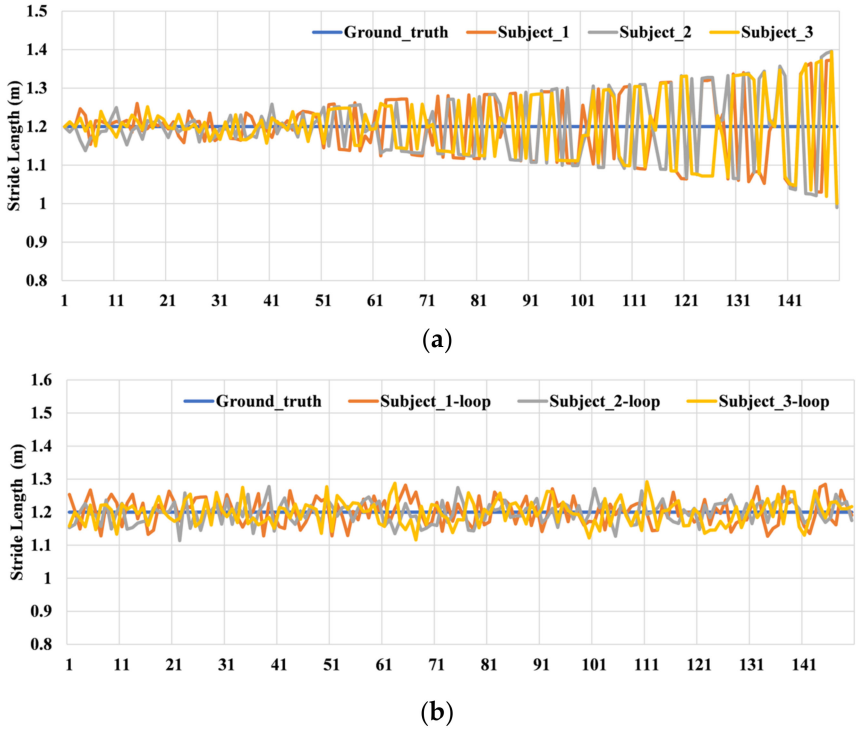


Figure 6. The stride-length estimation based on VIO without loop (**a**) and with loop detection (**b**).

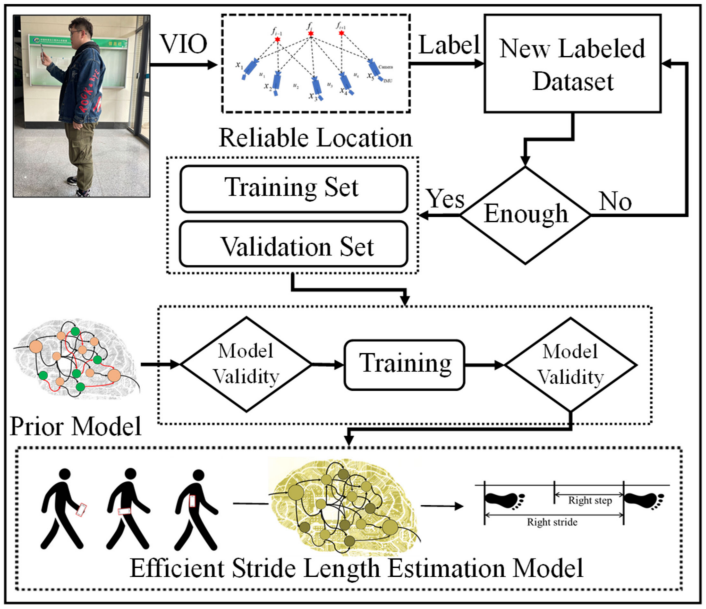


Figure 7. The framework of the updated SLE model based on online learning.

Moreover, we employ the VIO to the correct length of training data in this online learning module. The pre-correction mechanism obtains the ground truth. Specifically, the pedestrian holds a mobile phone with the camera open for a few seconds, as shown in the figure, and the visual and inertial data are from VIO and SLE, respectively. The results from the offline SLE model were compared with the step length from the visual-based method, and the model error D_{model} was obtained, as shown in the following:

$$D_{model} = \frac{1}{N} \cdot \sum_{i=1}^{N} (EL_i - L_i^{VIO}) \tag{18}$$

where N is the stride number, EL_i and L_i^{VIO} denote the predicted stride length and the ground-truth from VIO aiding of i^{th} stride.

When the difference is less than a particular threshold value, we believe the offline training model; otherwise, we use the new IMU data to update the model. The specific procedure for online learning presents in the following Algorithm 1:

Algorithm 1: The updating model procedure based on online learning

Input: IMU data, VIO-based locations, and offline SLE model
Output: a robust model for stride-length estimation
//Online test data obtained from a smartphone
IMU data segment according to the step detection algorithm, and remove the first segments
For each test segment **do**
    Calculating the ground truth of stride length from the statements based on VIO in Equation (16)
End for
Constructing the label data of each online segment as the form in Section 3.3
//Offline model verfication or updating
If the size of the online test data is enough **then**
    Evaluate the validity of the offline model according to Equation (18) by using online IMU data
    If difference $\geq$ threshold (low reliability of offline model) **then**
        Input the online test data as training data for the online learning framework and update the current SLE model
        Obtain the new online test data when keep walking and evaluate the validity of the updating model according to Equation (18)
        if difference < threshold (high reliability of offline model) **then**
            Return the updating model from the online learning
        Else
            Return the offline SLE model after validity with new test data
        End if
    End if
End if

In short, the online learning process can provide a pre-correction mechanism to reduce the offline model failure caused by personnel or equipment heterogeneity. At the same time, the online learning model is different from the offline model, which requires a large amount of training data. This online learning model only adjusts the CNN module parameters of the original model to adapt to the current users and output better step estimation results. Therefore, this process is a short learning process for learning new equipment or walking habits to improve the model's generalization in new scenarios.

4. Experimentation and Evaluation

In this section, we first present the implementation details of the efficient learning-based stride-length estimation model. Then, the experimental setup, including equipment and environments, is described. We conduct various tests on the datasets, including different pedestrians, devices, gaits, and environments, to evaluate the proposed learning-based SLE method's performance. The effectiveness of the proposed EL-SLE method is verified by comparing the localization results with the ground truth from the VIO-based localization. Last, we further apply our SLE model to the total distance and confirm the advantage of the proposed model.

4.1. Implementation

The efficient learning-based stride length estimation model was employed to provide reliable and robust pedestrian walking distance. Regarding the challenge of computation complexity, we employed client and server to calculate for model training and updating efficiently. A mobile application software designed in JAVA was implemented for Android

smartphones, and the sampling rates of MEMS and vision were 100 Hz and 50 Hz. As depicted in previous research [38,53], the temporal history of 200 IMU samples segmentation for each inference. The window length of 200 can explain the movement changes during one step, balancing the performance and computational load. Keras [57] with pandas and Adam [58], a first-order gradient-based approach, are used for data management and algorithm optimization in this paper. We gathered abundant moving characteristics inside data. To avoid overfitting, we adopted the Dropout way [59], randomly dropping 25% units from the neural network to lower the risk during training, which improves the generalization ability.

4.2. Experimental Setup

We conducted experiments in complex and changing scenes, including a gymnasium, sidewalk, playground, and underground shopping mall (the campus of the China University and Mining and Technology), which covers indoor and outdoor scenes, as shown in Figure 8.

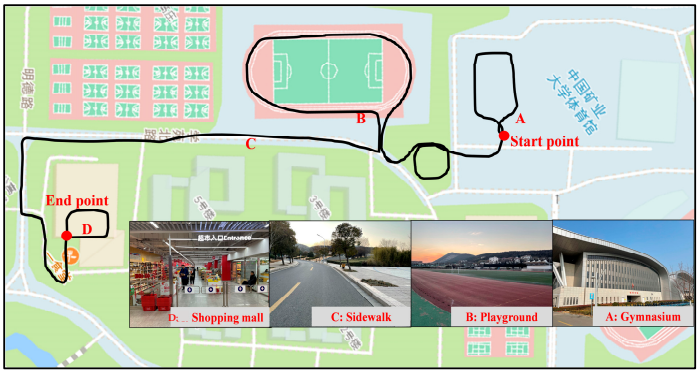


Figure 8. Pedestrian trajectory and test scenes description.

In the experiments, we used five Android smartphones, including four as test devices for IMU data collection and one as a tracking device for VIO; Figure 9 presents the scenario of the pedestrian collecting data. We utilized accurate visual-inertial odometry to obtain precise position information. For convenient comparison, we set Google Pixel XL 3 as the tracking device in the process, and the other four smartphones were set as the test devices. Five different users attached these devices to their body for data collection to reflect pedestrians moving in real life, and three gaits (slow walking, normal walking, and quick walking) were present during the process. Table 1 details the profiles of testers and smartphones. Our dataset has 20 sequences; we selected 15 sequences as training and the other 5 as tests. Table 2 presents the detail of sequences for the test. Our dataset's total walking distance and recording process is over 31.5 km and 8.1 h, which can cover pedestrian movement distance in daily life.

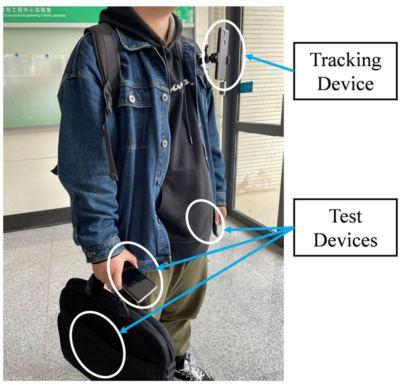


Figure 9. The scenario of data collection with different devices.

Table 1. Description of testers and devices.

	Abbreviation	Gender	Age	Height (cm)
Testers	T1	Male	26	178
	T2	Male	27	174
	T3	Female	23	162
	T4	Female	25	165
	T5	Male	29	181
	Abbreviation	**Model**	**Version**	**Inertial sensor**
Devices	D1	Google Pixel XL 3	9.0	BMI160
	D2	Samsung S8	8.0	LSM6DSL
	D3	Huawei Mate 30	11.5	ICM-20690
	D4	OPPO Reno 6	11.3	ICM-40607
	D5	Xiaomi 12	12.0	AK09918

Table 2. Details of the test sequences in the dataset.

Sequence	User (Gender)	Device	Attachments	Time (s)	Distance (m)
# 1	# 1/M	Samsung S8	Handheld	1495	1578
# 7	# 2/M	Huawei Mate 30	Handbag	1497	1573
# 11	# 3/F	OPPO Reno 6	Handbag	1450	1570
# 14	# 4/F	Xiaomi 12	Pocket	1462	1586
# 17	# 5/M	Samsung S8	Handheld	1457	1583

4.3. Evaluation Metrics

We employ the stride-length error rate and walking-distance error rate to evaluate the proposed method. Furthermore, the localization errors are also used as an evaluation index according to the PDR mechanism. The stride-length error rate and walking-distance error rate are calculated by the following:

$$\begin{aligned} E_s &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|L_e^i - L_g^i|}{L_g^i}\cdot 100\%\right) \\ E_{cd} &= \frac{\left|\sum_{i=1}^{M} L_e^i - \sum_{i=1}^{M} L_g^i\right|}{\sum_{i=1}^{M} L_g^i}\cdot 100\% \end{aligned} \tag{19}$$

where N represents the number of pedestrians walking, L_e^i and L_e^i represent the estimated length and the ground-truth of the i^{th} stride, respectively.

The following equation calculates the PDR-based localization error with different stride-length estimation models:

$$\begin{cases} x_i = x_{i-1} + L_e^i \cos\theta_e^i \\ y_i = y_{i-1} + L_e^i \sin\theta_e^i \\ e^i = \sqrt{(x_i - x_{VIO})^2 + (y_i - y_{VIO})^2} \end{cases} \tag{20}$$

where (x_i, y_i) denote the localization results from PDR. L_e^i and θ_e^i denote the estimated stride length and heading estimation at i^{th} stride, respectively. (x_{VIO}, y_{VIO}) represent the precise position results from VIO.

4.4. Performance of Stride-Length Estimation Model

4.4.1. Effect of Denoising Autoencoders and Preintegration

In this section, we analyze the effect of the data preprocessing, including data augmentation and preintegration, which is essential to obtain a reliable deep neural network for the SLE model. In the experiment, we compare the three stride-length estimation results:

(1) the stride length results from the direct learning-based stride-length estimation model (marked as "L-SLE"); (2) the results from the learning-based stride-length estimation with data augmentation (marked as "L-SLE + DA"); and (3) the results from the learning based stride-length estimation with data augmentation and preintegration (marked as "L-SLE + DA + PI"). The three models are used to train the neural networks with the dataset, and the results are obtained from each neural network model. Table 3 illustrates the performance of the results from the three models. We find that the stride-length estimation results from the preprocessing, including "L-SLE + DA" and "L-SLE + DA + PI", are better than the results from the direct learning manner, "L-SLE". Compared with results from "L-SLE", the mean error of stride length estimation results is reduced by 5% when using "L-SLE + DA" and "L-SLE + DA + PI".

Table 3. Effect of data preprocessing.

Test Seq.	L-SLE		L-SLE + DA		L-SLE + DA + PI	
	Mean Error (m)	Error Rate	Mean Error (m)	Error Rate	Mean Error (m)	Error Rate
# 1	0.067	4.32%	0.063	4.30%	0.063	4.29%
# 7	0.065	4.30%	0.062	4.29%	0.060	4.28%
# 11	0.059	4.28%	0.057	4.25%	0.054	4.24%
# 14	0.054	4.26%	0.053	4.23%	0.052	4.22%
# 17	0.062	4.29%	0.057	4.26%	0.058	4.26%
Ave.	0.061	4.29%	0.058	4.27%	0.057	4.23%

Meanwhile, the error rates from "L-SLE + DA" and "L-SLE + DA + PI" are 0.5% and 1.4%, and their results are less than that of "L-SLE". Therefore, the accuracy of SLE can be improved when the raw IMU data is with augmentation or preintegration. In addition, it is evident that "L-SLE + DA + PI" is better than "L-SLE + DA" in both mean error and error rate, and the results show that the average mean error reduces from 5.8 cm to 5.7 cm, and the average error rate from 4.27% to 4.23%. Concurrently, compared with "L-SLE + DA", "L-SLE + DA + PI" has less resource consumption to train for the neural network, reducing time consumption by about 0.15 ms/step because of the compact motion features vector by using preintegration processing.

4.4.2. Effect of the Adaptive Learning

The effect of the adaptive learning we proposed is analyzed in this section. The other two single neural networks, LSTM and CNN, are used to compare with the proposed adaptive learning. The two networks based on LSTM and CNN are the same as the structure of the neural network proposed in Section 3.2. These compared methods depend on a single neural network as the framework in previous studies [50,60]. We use the dataset to train the three models, and five test sequences are used to test their performance. The performance of the different networks for stride-length estimation is shown in Table 4. We find that the LSTM + CNN have the best performance in both max error and error rate. Compared to the other single networks, LSTM and CNN, the average max stride-length estimation error decreased by 4% and 25%, respectively, and the proposed network can significantly decrease the average error rate of stride-length estimation by nearly 5% and 10%.

Meanwhile, the results from the LSTM-based SLE are better than that from CNN; the reason is due to the advantage of RNN, which can efficiently extract the temporal feature to remember the feature relationship with time. However, CNN is suitable for extracting the features for the recognition task. Considering the random changes in pedestrian walking modes and scenes, we proposed adaptive learning to extract multiple features with the CNN and LSTM modules.

Table 4. The performance of the different neural networks.

Test Seq.	LSTM		CNN		LSTM + CNN	
	MAX Error (m)	Error Rate	MAX Error (m)	Error Rate	MAX Error (m)	Error Rate
# 1	0.334	4.47%	0.428	4.76%	0.326	4.29%
# 7	0.327	4.46%	0.426	4.74%	0.320	4.28%
# 11	0.328	4.43%	0.419	4.71%	0.314	4.24%
# 14	0.327	4.47%	0.422	4.68%	0.309	4.22%
# 17	0.331	4.51%	0.431	4.75%	0.316	4.26%
Ave.	0.329	4.47%	0.425	4.73%	0.317	4.26%

In the experiment, according to sequences #7 and #11, we analyze the stride length estimation results of LSTM, CNN, and LSTM + CNN in various walking modes and scenes. Figure 10 presents the box-plot of stride-length estimation error in various walking modes, including slow, normal, and quick walking. Furthermore, the qualitative results of three networks on sequences #7 and #11 are illustrated in Figure 11. It is clear that LSTM + CNN shows the best performance, and the CNN-based stride length method has the least accuracy, especially on sequence #11 with quick-walking mode. Meanwhile, LSTM + CNN is more stable than the other methods, and it can provide an excellent performance of stride-length estimation regardless of male and female users because of the multiple motion feature extraction. Based on the qualitative results in Figure 11, it is evident that the LSTM + CNN is close to the ground truth for sequences #7 and #11 in different scenes, while the CNN model's estimations are relatively more jittery on sequence #7. Therefore, the proposed LSTM + CNN model can provide accurate and stable stride-length estimation in various scenes compared with single networks.

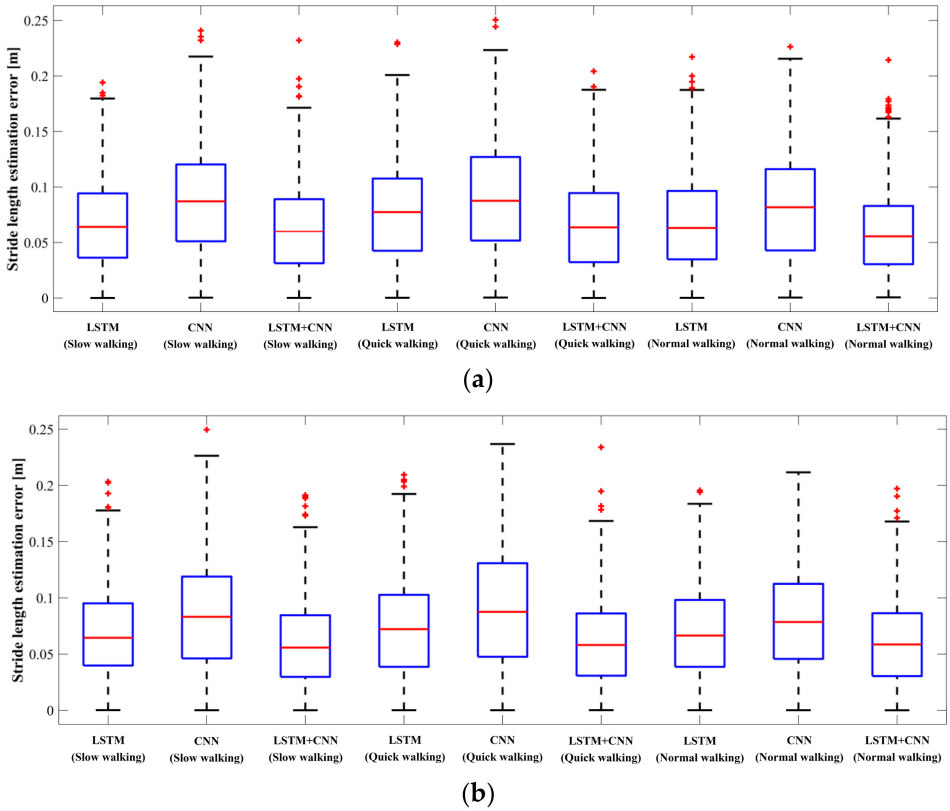


Figure 10. Box-plot of stride-length estimation error in three walking modes (slow walking, normal walking, and quick walking) with sequences #7 (**a**) and #11 (**b**).

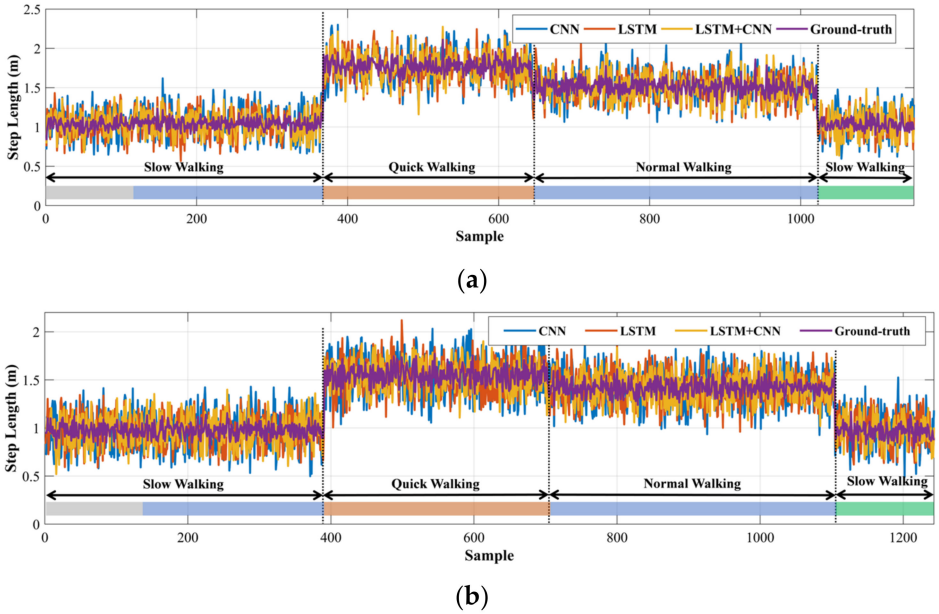


Figure 11. Qualitative comparison of the stride-length estimation results with a single network, CNN and LSTM, and a combined network, LSTM + CNN, on the sequence #7 (**a**) and #11 (**b**) in different scenes (indoor stadium, sidewalk, playground, and supermarket, are represented by color strip corresponding to gray, blue, orange, and green, respectively).

4.4.3. Comparison between Normal and Updated Models

In this section, we experiment to verify the effect of the online learning module in the proposed SLE model in Section 3.4. The new test sequences, sequences #21, #22, and #23, are collected from the new smartphones held by a new tester when walking on the same path, so we call these new IMU data unseen data, and the scenario of new data collection is shown in Figure 12. The unseen data we collect is used to test the generalization of the SLE model. For convenience, we kept smartphones facing forward to collect online labeled data for a few seconds. The process revises the SLE using the online learning module illustrated in Section 3.4. Table 5 summarizes the offline and updated models' stride-length estimation results. The updated model has better performance than the offline model. Compared with the offline model, the mean error decreases from 6.8 cm to 5.9 cm on average, and the mean error rate decreases from 4.33% to 4.26%.



Figure 12. The new sequences collection using the holder.

Meanwhile, the qualitative comparison of the offline SLE results and updated results on sequences #21, #22, and #23 are shown in Figures 13 and 14. We find that the stride-length accuracy is significantly improved by the updated model, making the more than 80% samples' accuracy error less than 0.1 m. The updated model results are better than the offline model, and it is robust to the different devices.

Table 5. Accuracy comparison of the offline model (SLE model without online learning module) and updated model (SLE model with online learning module).

Test Seq.	Tester	Device (IMU)	Offline Model		Updated Model	
			Mean Error (m)	Error Rate	Mean Error (m)	Error Rate
# 21	T #6	Vivo X70 (LSM6DSO)	0.068	4.33%	0.057	4.25%
# 22	T #6	Xiaomi 10 (LSM6DS0)	0.066	4.31%	0.055	4.24%
# 23	T #6	Galaxy F52 (MMC5603)	0.072	4.37%	0.064	4.29%
Ave.			0.068	4.34%	0.059	4.26%

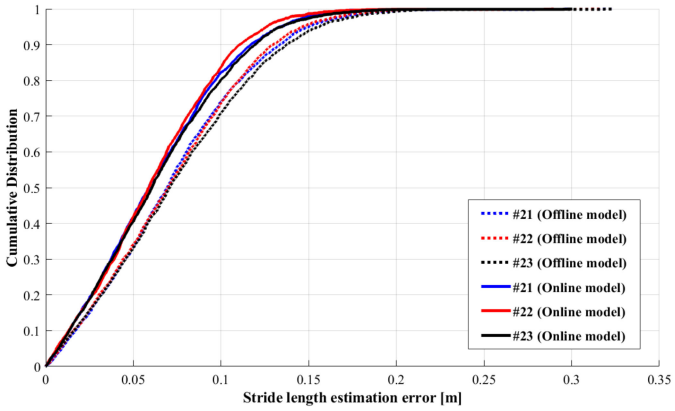


Figure 13. Qualitative comparison of stride-length estimation error between the offline model (SLE model without online learning module) and updated model (SLE model with online learning module).

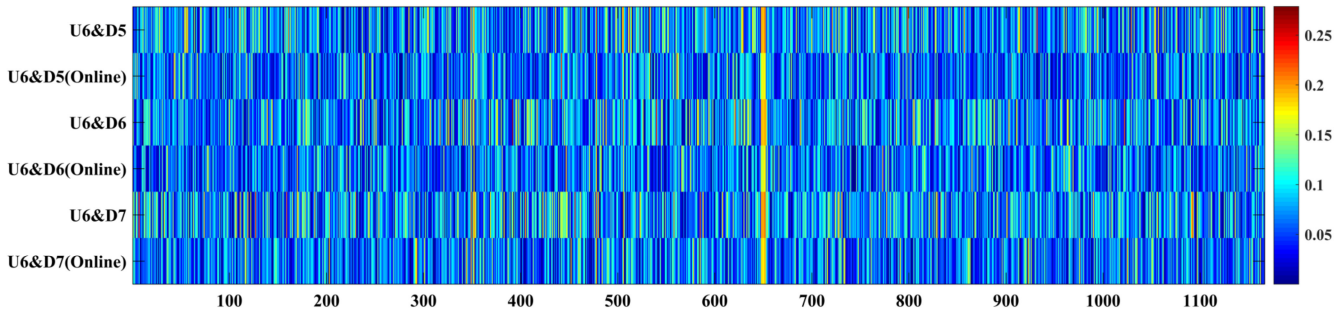


Figure 14. The stride-length errors of the offline model (SLE model without online learning module) and updated model (SLE model with online learning module) with different devices.

4.5. Walking-Distance Estimation

This section analyzes the SLE models' cumulative walking distance estimation to evaluate the overall performance and robustness in the long-term walking scenario. The cumulative walking distance is from the number of strides and the corresponding stride-length estimation results. The walking distance is calculated as follows:

$$L_{WD} = \sum_{i=1}^{N} L_i \tag{21}$$

where N is the number of strides and L_i represents the SLE results of the i^{th} stride.

In the experiment, the dataset we applied to train models is the same as the data in Section 4.2, and the test sequences, including three additional new sequences, #21, #22, and #23, are used to analyze the performance of the walking distance estimation. Table 6 shows the error and error rate of the stride length estimation with three SLE models. It is easy to

see that adaptive learning with preprocessing and online learning (marked as "Adaptive learning + PP + OL") can significantly decrease the error and error rate, especially on the new sequences #21, #22, and #23. The reason is that the online learning module can efficiently optimize the parameter of the CNN module. Compared with direct adaptive learning and "Adaptive learning + PP", "Adaptive learning + PP + OL" reduces the average error by nearly 8% and 4%, respectively, and decreases the error rate of walking distance by 0.3% and 0.2% on average, respectively. In terms of robustness of the model, we find that "Adaptive learning + PP + OL" is relatively stable, and the max error of walking distance from the online learning-based model is 45.59 m, which is significantly less than the other two models. The findings benefit from the outstanding online learning module of the proposed model, which mitigates the heterogeneity when the motion data from new testers or devices.

Table 6. Performance comparison of walking distance with different models, including Adaptive learning, Adaptive learning with Preprocessing (Adaptive learning + PP), and Adaptive learning with Preprocessing and Online Learning (Adaptive learning + PP + OL).

Model	Attributes	Seq#1	Seq#7	Seq#11	Seq#14	Seq#17	Seq#21	Seq#22	Seq#23	Ave.
Adaptive learning	Error (m)	45.62	42.37	39.31	37.97	42.68	50.97	51.36	51.08	45.17
	Error rate (%)	2.9	2.7	2.5	2.4	2.7	3.2	3.3	3.1	2.9
Adaptive learning + PP	Error (m)	43.75	40.55	39.15	36.51	40.09	48.28	49.96	50.77	43.63
	Error rate (%)	2.8	2.6	2.5	2.3	2.5	3.1	3.1	3.2	2.8
Adaptive learning + PP + OL	Error (m)	43.63	39.43	39.24	35.94	39.81	45.57	44.38	45.59	41.70
	Error rate (%)	2.8	2.5	2.5	2.3	2.5	2.8	2.8	2.9	2.6

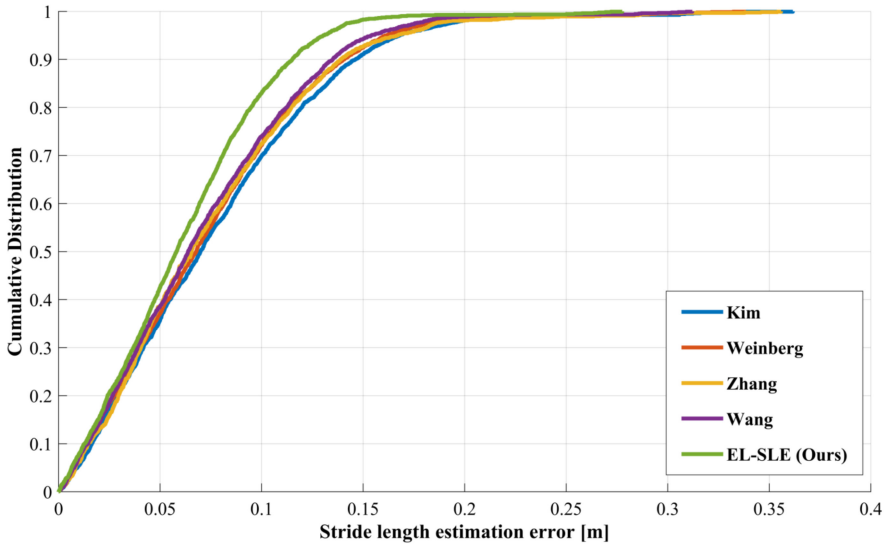
4.6. Comparison with Other Methods

So far, the above experiments have been conducted to demonstrate the performance of each module in the proposed model for stride-length estimation. To verify the superiority of the proposed EL-SLE model, we compare the proposed method with four established methods in terms of the error rate of stride length. The first is that proposed by Kim, who proposed an empirical model according to the average acceleration magnitude during pedestrian walking [8]; the second is from Weinberg, who proposed a conventional SLE model based on the component strategy with vertical acceleration in each pedestrian walking stride [7]; the SLE model based on recurrent neural network is proposed to solve the problem, which is used as the third compared method [50]; and the fourth compared method is based on the pedestrian mode recognition for accurate stride-length estimation during complex walking scenarios [34].

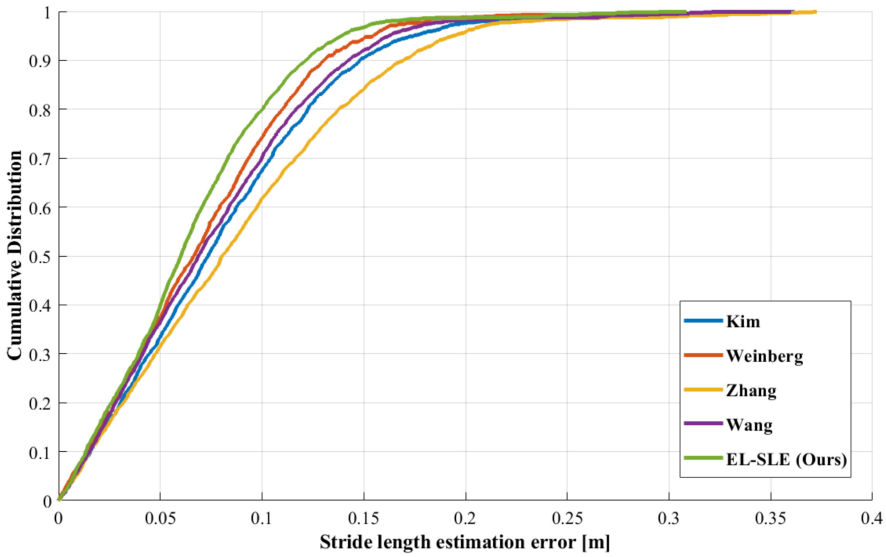
To conduct the exhaustive experiments, we use the eight test sequences, including three unseen data, sequences #21, #22, and #23, which is challenging for learning-based methods. In the experiment, the error rate and absolute errors of stride length are used to verify the performance of SLE models. At the same time, we analyze the end points' positioning error by using the standard PDR algorithm, which is a significant and practical index to evaluate performance. In the PDR mechanism, the models have the same heading angles, and the angles are provided by the ground truth from VIO in Section 3.3. The error rate and end points' positioning error are shown in Table 7, and Figure 15 illustrates the cumulative distribution diagram of absolute errors of stride-length estimation. In the table, we find that the Zhang model has the worst results among these methods, and the error rate of stride length is 5.83% on #23.

Table 7. Performance of stride-length estimation with different SLE models.

Methods	Attributes	Seq #1	Seq #7	Seq #11	Seq #14	Seq #17	Seq #21	Seq #22	Seq #23	Ave.
Kim [8]	Error rate (%)	5.35	5.34	4.74	4.70	5.29	5.33	5.31	5.38	5.18
	Positioning error (m)	17.80	17.62	16.68	16.92	17.35	18.16	17.07	17.83	17.43
Weinberg [7]	Error rate (%)	5.03	4.96	4.77	4.70	5.16	5.09	5.17	5.11	5.00
	Positioning error (m)	16.33	15.75	14.97	14.81	16.56	16.38	16.75	17.92	16.18
Zhang [50]	Error rate (%)	5.36	5.23	5.16	5.08	5.54	5.78	5.72	5.83	5.46
	Positioning error (m)	17.82	17.14	17.09	16.97	18.42	18.89	18.75	19.01	18.01
Wang [34]	Error rate (%)	4.57	4.42	4.39	4.36	4.53	4.52	4.56	4.49	4.48
	Positioning error (m)	13.14	12.79	12.37	12.27	13.05	13.10	13.22	13.08	12.88
EL-SLE (ours)	Error rate (%)	4.31	4.28	4.25	4.21	4.26	4.25	4.24	4.29	4.26
	Positioning error (m)	8.54	8.37	7.92	7.63	8.83	8.97	7.80	9.03	8.39



(**a**)



(**b**)

Figure 15. Cumulative distribution diagram of absolute errors of stride-length estimation with different SLE models on sequences #1 (**a**) and #21 (**b**).

Meanwhile, the end point's positioning error of sequence #23 is more than 18 m, so the errors in stride-length estimation can lead to severe positioning deviation in the PDR

mechanism. The reason is that the LSTM-based SLE model has poor generalization ability on new devices or pedestrians. On the contrary, the proposed method shows excellent performance in all sequences. The average error rate of the proposed method is 4.26%, and the proposed model obtained 8.39 m of end point's positioning error on average. The results are thanks to the online learning module of our model, which can extract the new CNN-based features for mode recognition. Compared with Kim, Weinberg, Tapeline, and Wang, the average end points' positioning errors were reduced by nearly 51%, 48%, 53%, and 39%, respectively. Furthermore, we find that the proposed EL-SLE model shows more stability than the other four models, and the error rate of stride-length estimation is less than 4.4%. Therefore, our model provides more accurate and robust results of stride-length estimation during complex walking modes and dynamic scenes, and the accurate results can significantly reduce positioning errors in location-based services. Figure 15 shows the cumulative distribution diagram of absolute errors of stride length estimation with different SLE models on sequences #1 and #21. The figure clearly shows that the proposed method performs better than the other methods on the selected sequences, and the performance gap between the proposed model and other models is even more evident in sequence #21. For the unseen data, the Kim, Weinberg, and Wang models can obtain more accurate results than the Zhang model, which depends on the direct LSTM. The reason is that the simple LSTM-based SLE model can only extract the temporal features in the fixed mode during walking. Nevertheless, the proposed method employed LSTM and CNN to extract the multiple motion features, providing reasonable assistance for walking mode recognition during long-term and complex movement. In addition, the parameters of CNN are optimized to process the unseen data using an online learning module. Therefore, the proposed model performs better than other models, such as the empirical, pattern recognition, and direct LSTM methods.

5. Conclusions

This paper proposes a learning-based stride-length estimation method to approach the challenge of device or user heterogeneity for the SLE model by using an adaptive learning module. We are the first to propose the adaptive learning strategy using LSTM and CNN to extract both temporal characteristics and statistical features from IMU data, strengthening the model's robustness under challenging conditions. Additionally, we use denoising autoencoders and preintegration for data preparation, improving the training efficiency of neural networks. The output of two network modules is then combined with mapping the eigenvectors of the stride length. Furthermore, the online learning module is proposed to extract new features and optimize CNN parameters for the customized model. Meanwhile, we suggested a simple method that employs the built-in sensors of smartphones and requires no additional hardware, increasing the convenience of labeled data for network training. Extensive trials in challenging indoor and outdoor environments are conducted to assess the viability of the proposed EL-SLE. The experimental findings support the generalizability of the SLE model, and the results demonstrate that our approach is capable of superior stride-length estimates than the other models. Our method's average stride-length error rate is 4.26%, significantly less than the state-of-the-art SLE methods. We analyze the application of SLE for PDR-based pedestrian positioning, and the results show that the proposed method can obtain more accurate positioning in long-term walking. Compared with other methods, EL-SLE decreases the average positioning error of endpoints by nearly 51%, 48%, 53%, and 39%, respectively. In addition to reducing the positioning error for PDR, the accurate stride-length estimation can be employed to provide better service for sports analytics, health care, and extensive data statistics for the Internet of Things.

However, some limitations need to be approached in our future work. The proposed model has been verified with experiments that involve normal walking, while the test gait does not include more special motion gait such as backing, side walking, and jumping. Moreover, the proposed model needs to be processed by a portable computer during online

learning, and the process takes a little time when the data is transmitted between the mobile devices and the server, leading to delay problems during step estimation and the obstacle of practical application. In the future, the stride-length estimation model needs to be extended to test with more complex gaits and dynamic environments, and we will consider 5G and signal compression transmission technology to achieve low-delay result output in the Internet of Things.

Author Contributions: Conceptualization, M.S., G.C. and Z.Z.; methodology, M.S.; software, Z.Z.; validation, M.S.; formal analysis, G.C.; investigation, Z.Z.; resources, M.S.; data curation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, M.S. and G.C.; visualization, M.S.; supervision, G.C.; project administration, G.C. and Z.Z.; funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Training Program of the Major Research Plan of the National Natural Science Foundation of China (Grant No. 2021ZDPY0220), a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and a project funded by Jiangsu Province Science and Technology Planning Project under grant BE2022716.

Acknowledgments: We would like to thank the School of Environment Science and Spatial Informatics for its support in our research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Harwahyu, R.; Cheng, R.-G.; Wei, C.-H.; Sari, R.F. Optimization of Random Access Channel in NB-IoT. *IEEE Internet Things J.* **2018**, *5*, 391–402. [CrossRef]
2. Jylhä, M.; Guralnik, J.M.; Balfour, J.L.; Fried, L.P. Walking difficulty, walking speed, and age as predictors of self-rated health: The women's health and aging study. *J. Gerontol. Ser. A Biol. Sci. Med Sci.* **2001**, *56*, M609–M617. [CrossRef] [PubMed]
3. Woo, J.; Ho, S.C.; Yu, A.L. Walking Speed and Stride Length Predicts 36 Months Dependency, Mortality, and Institutionalization in Chinese Aged 70 and Older. *J. Am. Geriatr. Soc.* **1999**, *47*, 1257–1260. [CrossRef] [PubMed]
4. Yu, N.; Zhan, X.; Zhao, S.; Wu, Y.; Feng, R. A Precise Dead Reckoning Algorithm Based on Bluetooth and Multiple Sensors. *IEEE Internet Things J.* **2018**, *5*, 336–351. [CrossRef]
5. Kuang, J.; Niu, X.; Chen, X. Robust Pedestrian Dead Reckoning Based on MEMS-IMU for Smartphones. *Sensors* **2018**, *18*, 1391. [CrossRef]
6. Hostettler, R.; Särkkä, S. IMU and magnetometer modeling for smartphone-based PDR. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain, 4–7 October 2016; pp. 1–8.
7. Weinberg, H. Using the ADXL202 in pedometer and personal navigation applications. *Analog Devices AN-602 Appl. Note* **2002**, *2*, 1–6.
8. Kim, J.W.; Jang, H.J.; Hwang, D.-H.; Park, C. A Step, Stride and Heading Determination for the Pedestrian Navigation System. *J. Glob. Position. Syst.* **2004**, *3*, 273–279. [CrossRef]
9. Yao, Y.; Pan, L.; Fen, W.; Xu, X.; Liang, X.; Xu, X. A Robust Step Detection and Stride Length Estimation for Pedestrian Dead Reckoning Using a Smartphone. *IEEE Sens. J.* **2020**, *20*, 9685–9697. [CrossRef]
10. Shavit, Y.; Ferens, R.; Keller, Y. Paying Attention to Activation Maps in Camera Pose Regression. *arXiv* **2021**, arXiv:2103.11477.
11. Nigg, B.M.; Bahlsen, H.A.; Luethi, S.M.; Stokes, S. The influence of running velocity and midsole hardness on external impact forces in heel-toe running. *J. Biomech.* **1987**, *20*, 951–959. [CrossRef]
12. Brughelli, M.E.; Cronin, J.B.; Chaouachi, A. Effects of Running Velocity on Running Kinetics and Kinematics. *J. Strength Cond. Res.* **2011**, *25*, 933–939. [CrossRef]
13. Lee, S.-W.; Mase, K.; Kogure, K. Detection of Spatio-Temporal Gait Parameters by Using Wearable Motion Sensors. In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 17–18 January 2006; pp. 6836–6839.
14. Clark, R.A.; Bower, K.J.; Mentiplay, B.F.; Paterson, K.L.; Pua, Y.-H. Concurrent validity of the Microsoft Kinect for assessment of spatiotemporal gait variables. *J. Biomech.* **2013**, *46*, 2722–2725. [CrossRef]
15. Albert, J.A.; Owolabi, V.; Gebel, A.; Brahms, C.M.; Granacher, U.; Arnrich, B. Evaluation of the Pose Tracking Performance of the Azure Kinect and Kinect v2 for Gait Analysis in Comparison with a Gold Standard: A Pilot Study. *Sensors* **2020**, *20*, 5104. [CrossRef]
16. Webster, K.E.; Wittwer, J.E.; Feller, J.A. Validity of the GAITRite walkway system for the measurement of averaged and individual step parameters of gait. *Gait Posture* **2005**, *22*, 317–321. [CrossRef]
17. Díez, L.E.; Bahillo, A.; Otegui, J.; Otim, T. Step Length Estimation Methods Based on Inertial Sensors: A Review. *IEEE Sens. J.* **2018**, *18*, 6908–6926. [CrossRef]
18. Foxlin, E. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comput. Graph. Appl.* **2005**, *25*, 38–46. [CrossRef]

19. Qiu, S.; Wang, Z.; Zhao, H.; Qin, K.; Li, Z.; Hu, H. Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion. *Inf. Fusion* **2018**, *39*, 108–119. [CrossRef]
20. Zhang, W.; Wei, D.; Yuan, H. The Improved Constraint Methods for Foot-Mounted PDR System. *IEEE Access* **2020**, *8*, 31764–31779. [CrossRef]
21. Lan, K.-C.; Shih, W.-Y. On Calibrating the Sensor Errors of a PDR-Based Indoor Localization System. *Sensors* **2013**, *13*, 4781–4810. [CrossRef]
22. Do, T.-N.; Liu, R.; Yuen, C.; Zhang, M.; Tan, U.-X. Personal Dead Reckoning Using IMU Mounted on Upper Torso and Inverted Pendulum Model. *IEEE Sens. J.* **2016**, *16*, 7600–7608. [CrossRef]
23. Guo, Y.; Li, Y.; Sun, Y.L. Accurate indoor localization based on crowd sensing. In Proceedings of the 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, Cyprus, 5–9 September 2016; pp. 708–713.
24. Wang, A.; Ou, X.; Wang, B. Improved Step Detection and Step Length Estimation Based on Pedestrian Dead Reckoning. In Proceedings of the 2019 IEEE 6th International Symposium on Electromagnetic Compatibility (ISEMC), Nanjing, China, 1–4 November 2019; pp. 1–4.
25. Álvarez, D.; González, R.C.; López, A.M.; Álvarez, J.C. Comparison of Step Length Estimators from Weareable Accelerometer Devices. In Proceedings of the 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, USA, 30 August–3 September 2006; pp. 5964–5967.
26. Zhang, P.; Chen, X.; Ma, X.; Wu, Y.; Jiang, H.; Fang, D.; Tang, Z.; Ma, Y. SmartMTra: Robust Indoor Trajectory Tracing Using Smartphones. *IEEE Sens. J.* **2017**, *17*, 3613–3624. [CrossRef]
27. Xu, S.; Chen, R.; Yu, Y.; Guo, G.; Huang, L. Locating Smartphones Indoors Using Built-In Sensors and Wi-Fi Ranging with an Enhanced Particle Filter. *IEEE Access* **2019**, *7*, 95140–95153. [CrossRef]
28. Diaz, E.M.; Gonzalez, A.L.M. Step detector and step length estimator for an inertial pocket navigation system. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 105–110.
29. Bousdar Ahmed, D.; Munoz Diaz, E.; García Domínguez, J.J. Automatic Calibration of the Step Length Model of a Pocket INS by Means of a Foot Inertial Sensor. *Sensors* **2020**, *20*, 2083. [CrossRef]
30. Martinelli, A.; Gao, H.; Groves, P.D.; Morosi, S. Probabilistic Context-Aware Step Length Estimation for Pedestrian Dead Reckoning. *IEEE Sens. J.* **2018**, *18*, 1600–1611. [CrossRef]
31. Ju, H.; Park, S.Y.; Park, C.G. A Smartphone-Based Pedestrian Dead Reckoning System with Multiple Virtual Tracking for Indoor Navigation. *IEEE Sens. J.* **2018**, *18*, 6756–6764. [CrossRef]
32. Huang, B.; Qi, G.; Yang, X.; Zhao, L.; Zou, H. Exploiting cyclic features of walking for pedestrian dead reckoning with unconstrained smartphones. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 374–385.
33. Gao, H.; Groves, P.D. Context Determination for Adaptive Navigation using Multiple Sensors on a Smartphone. In Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016), Portland, OR, USA, 12–16 September 2016.
34. Wang, Q.; Ye, L.; Luo, H.; Men, A.; Zhao, F.; Ou, C. Pedestrian Walking Distance Estimation Based on Smartphone Mode Recognition. *Remote. Sens.* **2019**, *11*, 1140. [CrossRef]
35. Diaz, E.M. Inertial Pocket Navigation System: Unaided 3D Positioning. *Sensors* **2015**, *15*, 9156–9178. [CrossRef]
36. Hannink, J.; Kautz, T.; Pasluosta, C.F.; Barth, J.; Schülein, S.; Gaßmann, K.-G.; Klucken, J.; Eskofier, B.M. Mobile Stride Length Estimation with Deep Convolutional Neural Networks. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 354–362. [CrossRef]
37. Pham, T.T.; Suh, Y.S. Walking Step Length Estimation Using Waist-Mounted Inertial Sensors with Known Total Walking Distance. *IEEE Access* **2021**, *9*, 85476–85487. [CrossRef]
38. Chen, C.; Zhao, P.; Lu, C.X.; Wang, W.; Markham, A.; Trigoni, A. Deep-Learning-Based Pedestrian Inertial Navigation: Methods, Data Set, and On-Device Inference. *IEEE Internet Things J.* **2020**, *7*, 4431–4441. [CrossRef]
39. Sui, J.-D.; Chang, T.-S. IMU Based Deep Stride Length Estimation with Self-Supervised Learning. *IEEE Sens. J.* **2021**, *21*, 7380–7387. [CrossRef]
40. Wang, Q.; Luo, H.; Ye, L.; Men, A.; Zhao, F.; Huang, Y.; Ou, C. Personalized Stride-Length Estimation Based on Active Online Learning. *IEEE Internet Things J.* **2020**, *7*, 4885–4897. [CrossRef]
41. Fusco, G.; Coughlan, J.M. Indoor Localization Using Computer Vision and Visual-Inertial Odometry. In Computers Helping People with Special Needs. ICCHP 2018. Lecture Notes in Computer Science 2018. *Comput. Help. People Spec. Needs* **2018**, *10897*, 86–93.
42. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
43. Abolfazli Esfahani, M.; Wang, H.; Wu, K.; Yuan, S. AbolDeepIO: A Novel Deep Inertial Odometry Network for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1941–1950. [CrossRef]
44. Abadi, M.J.; Luceri, L.; Hassan, M.; Chou, C.T.; Nicoli, M. A collaborative approach to heading estimation for smartphone-based PDR indoor localisation. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 554–563.

45. Mohammed, S.; Tashev, I. Unsupervised deep representation learning to remove motion artifacts in free-mode body sensor networks. In Proceedings of the 2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Eindhoven, The Netherlands, 9–12 May 2017; pp. 183–188.
46. Liolios, C.; Doukas, C.; Fourlas, G.K.; Maglogiannis, I. An overview of body sensor networks in enabling pervasive healthcare and assistive environments. In Proceedings of the PETRA '10 the 3rd International Conference on PErvasive Technologies Related to Assistive Environments, Samos, Greece, 23–25 June 2010.
47. Qian, W.; Zhu, Y.; Jin, Y.; Yang, J.; Qi, P.; Wang, Y.; Ma, Y.; Ji, H. A Pedestrian Navigation Method Based on Construction of Adapted Virtual Inertial Measurement Unit Assisted by Gait Type Classification. *IEEE Sens. J.* **2021**, *21*, 15258–15268. [CrossRef]
48. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
49. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
50. Ping, Z.; Zhidong, M.; Pengyu, W.; Zhi-hong, D. Pedestrian Stride-Length Estimation Based on Bidirectional LSTM Network. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 3358–3363.
51. Chen, C.; Rosa, S.; Lu, C.X.; Trigoni, A.; Markham, A. SelectFusion: A Generic Framework to Selectively Learn Multisensory Fusion. *arXiv* **2019**, arXiv:1912.13077.
52. Han, Y.C.; Wong, K.I.; Murray, I. Stride Length Estimation Based on a Single Shank's Gyroscope. *IEEE Sens. Lett.* **2019**, *3*, 1–4. [CrossRef]
53. Yan, H.; Herath, S.; Furukawa, Y. RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, & New Methods. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3146–3152.
54. Herath, S. Robust Neural Inertial Navigation in the Wild. Master's Thesis, University of Moratuwa, Applied Sciences, School of Computing Science, Moratuwa, Sri Lanka, 2019.
55. Jin, Y.; Toh, H.-S.; Soh, W.-S.; Wong, L.W.-C. A robust dead-reckoning pedestrian tracking system with low cost sensors. In Proceedings of the 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom), Seattle, WA, USA, 21–25 March 2011; pp. 222–230.
56. Parodi, B.B.; Lenz, H.; Szabo, A.; Wang, H.; Horn, J.; Bamberger, J.; Obradovic, D. Initialization and Online-Learning of RSS Maps for Indoor/Campus Localization. In Proceedings of the 2006 IEEE/ION Position, Location, and Navigation Symposium, San Diego, CA, USA, 25–27 April 2006; pp. 164–172.
57. Ketkar, N. Introduction to keras. In *Deep Learning with Python*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 97–111.
58. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
59. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
60. Klein, I.; Asraf, O. "StepNe" Deep Learning Approaches for Step Length Estimation. *IEEE Access* **2020**, *8*, 85706–85713. [CrossRef]