Deep Learning Based Gait Recognition Using Smartphones in the Wild

Qin Zou, Yanling Wang, Qian Wang, Yi Zhao, Qingquan Li

***Abstract*—Comparing with other biometrics, gait has advantages of being unobtrusive and difficult to conceal. Inertial sensors such as accelerometer and gyroscope are often used to capture gait dynamics. Nowadays, these inertial sensors have commonly been integrated in smartphones and widely used by average person, which makes it very convenient and inexpensive to collect gait data. In this paper, we study gait recognition using smartphones in the wild. Unlike traditional methods that often require the person to walk along a specified road and/or at a normal walking speed, the proposed method collects inertial gait data under a condition of unconstraint without knowing when, where, and how the user walks. To obtain a high performance of person identification and authentication, deep-learning techniques are presented to learn and model the gait biometrics from the walking data. Specifically, a hybrid deep neural network is proposed for robust gait feature representation, where features in the space domain and in the time domain are successively abstracted by a convolutional neural network and a recurrent neural network. In the experiments, two datasets collected by smartphones on a total of 118 subjects are used for evaluations. Experiments show that the proposed method achieves over 93.5% and 93.7% accuracy in person identification and authentication, respectively.**

***Index Terms*—Gait recognition, inertial sensor, person identification, convolutional neural network, recurrent neural network.**

I. Introduction

BIOMETRICS refers to the automatic identification of a person based on his or her physiological or behavioral characteristics. With the increasing demands of person identification and verification in the era of big data and artificial intelligence, the research and development of biometric systems have attracted wide attention from both the academia and the industry. Up to date, many biometrics have been commercially used, e.g., the fingerprint, iris, face, and voice, etc. Among these biometrics, some are obtrusive to users as they require the cooperation of users to collect the data. For example, users are asked to place the finger on the device to have their fingerprints captured, or to look at the camera close enough to have their irises imaged. In such cases, the users may feel offended, and can easily realize that their identities are being checked. Meanwhile, some biometrics may easily be forged and attacked. For example, the face recognition can be cheated by using an image or a video storing the target face [1], [2]. As a result, less obtrusive and more robust biometrics are still in strong desire in the current stage.

A property of being unobtrusive is especially important for a biometric system that is required to work in a hidden way, e.g., to recognize the identity of a person but not to let him/her know he/she is being identified. Among various biometrics, gait is such a biometric that not only satisfies the requirement of being unobtrusive but also has the advantage of being difficult to conceal [3]–[5]. However, as behavioral biometrics with unobtrusive property are far more difficult to extract and are usually combined, it is a challenge to use them.

Gait biometric refers to identifying a person based on his/her walking manner. Generally, the gait recognition can be performed on two types of data. One is a sequence of silhouette images, and the other is an inertial gait time-series. In the scenario that we can capture the gait silhouette images or inertial gait time-series, we can perform gait recognition and hence person identification.

For silhouette image based methods, as the segmentation of a walking person from arbitrary background is still an unsolved open problem, it is only effective when performing gait recognition in some constrained environments [6], e.g., in a room or along a corridor with simple image background. While for inertia-based methods, inertial sensors such as accelerometer and gyroscope are used to record the inertial data generated by the movement of a walking body. These inertial data capture the gait dynamics in a general way and have been proofed to be useful for extracting the walking patterns [7]. In the past decade, a number of methods have been developed for inertia-based gait recognition [8]–[12]. However, most of these methods require the inertia sensors be fasten to some specific joints of the human body, which brings inconveniences for gait data collection.

Nowadays, smartphones have been widely used by average person, and many advanced inertial sensors including accelerometer and gyroscope have been commonly integrated into the smartphones. As a result, it is very convenient and inexpensive to collect inertia gait data, which inspired a number of methods to use smartphones for gait recognition [13]–[15]. Practical demands for smartphone-based gait recognition potentially come from two aspects. The first is person identification, e.g., it can help the government find a specific person such as a criminal. The second is person authentication, e.g., it can help automatically lock the lost phone when someone else gets it. In order to obtain a high accuracy of person identification, most existing methods require the person to walk along a specified road such as a straight lounge and/or at a normal walking speed. These constrains heavily limit the application. To loosen the constraints and expand the application scenario, more robust gait-recognition algorithms are demanded.

In recent years, deep learning has demonstrated state-of-the-art, human-competitive, and sometimes better-than-human performance in solving many cognitive problems such as speech recognition [16] and visual perception [17],

Q. Zou, Y. Wang, Q. Wang and Y. Zhao are with the School of Computer Science, Wuhan University, Wuhan 430072, P.R. China (E-mails: {qzou, yanling, yizhaowhu, qianwang}@whu.edu.cn).

Q. Li is with Shenzhen Key Laboratory of Spatial Smart Sensing and Service, Shenzhen University, Guangdong 518060, P.R. China (E-mail: liqq@szu.edu.cn).

[18]. There are mainly two types of deep neural networks. One is the deep convolutional neural network (DCNN), and the other is the deep recurrent neural network (DRNN). The former convolves the input signals in the space domain and is talent in handling two dimensional signals such as the images. The later processes the input signals in a recursive way, and is skilled in handling time-series such as the voices. For the data produced by accelerometer and gyroscope, they can be naturally arranged into a two-dimension time-series. Hence, the DCNNs can place a strength to represent the inertial data with convolutional feature maps, and the DRNNs can exert their advantages by processing them as time-series. However, how to combine the two types of neural networks for effective gait representation is lack of study.

In this paper, we study gait recognition using smartphones in the wild, and propose an effective and seamless combination of DCNN and DRNN for robust inertial gait feature representation. In gait data collection, the smartphones are assumed to be used in a condition of unconstrained, and they don't record when, where, and how the user walks. Under this assumption, firstly, the inertial data collected by smartphones is partitioned into the walking session and the non-walking session by a fully convolutional neural network, where hierarchical convolutional features are fused together to accurately extract the walking session. Then, gait features are extracted on the walking data by the proposed hybrid deep learning technique. Specifically, the three-dimension data (in X, Y and Z axis) of the accelerometer and the gyroscope are put together to form a six-dimension time-series. Then, a convolutional neural network with one-dimension kernels is designed to convolve the input time-series into convolutional feature maps, which can still hold the property of time-series. After that, the time-series convolutional features are processed by a recurrent neural network for robust gait feature extraction. Based on the above operations, person identification and authentication models can finally be constructed in a supervised training manner. The experiments on two datasets demonstrate the effectiveness of the proposed method in terms of person identification and authentication.

The main contributions of this paper lie in three-fold:

- First, for the problem that inertia-based gait recognition is only effective in restricted environments, a novel gait-recognition method using the smartphone as data collector in a condition of unconstraint is proposed. In this method, the inertial data are collected by smartphones in the living environment without any limit, which can largely expand the application scenario of gait recognition.
- Second, for the problem that traditional methods can hardly achieve high performance in inetia-based gait recognition, and DCNN and DRNN often function in different domains in separated ways, a novel deep-learning architecture is designed to seamlessly integrate the two networks for gait feature representation. Specifically, a DCNN with one-dimension kernels convolves the input time-series into convolutional feature maps, and then an LSTM (Long Short-Term Memory) network processes the resulted feature maps for gait feature extraction. The extracted features are found to be very discriminative for person identification and authentication.
- Third, two main datasets are collected for performance evaluation. One dataset collected on 20 subjects with each subject containing thousands of samples. The other dataset is collected on 118 subjects with each subject containing hundreds of samples. Based on the two datasets, six sub-datasets are constructed, which can be used for quantitative evaluation and performance comparison for different gait-recognition methods.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III describes the gait-data collection and preprocessing. Section IV introduces the proposed approach, including the network design and network integration. Section V reports the experiments and results. Section VI concludes our work.

II. Related Work

A. Gait Recognition Using Inertial Sensors

Sensor-based gait recognition can be performed in three ways: by sensors in the floor [19], by sensors in the shoes [20], and by sensors on the body [8]. Among these methods and their variations, inertia-sensor based methods are the most attractive. It is because that, the inertial sensors can be easily placed on the body to capture the details of the movement characteristics [21]–[25], and the captured time-series data are effective for gait-based person identification and authentication [12], [26].

In early research of inertia-based gait recognition, Ailisto *et al.* proposed a signal-correlation method, where the recognition was performed in means of template matching and cross-correlation computation [27], [28]. Following this work, Gafurov *et al.* made many significant improvements [8], [29]–[31]. In [30], they analyzed the minimal-effort impersonation attack and the closest person attack on gait biometrics. In [31], they collected 300 gait sequences from 50 subjects by placing an accelerometer sensor in user's pocket, and achieved an equal error rate (EER) of 7.3%. In [8], they tried foot-, pocket-, arm- and hip-based user authentication and found that a sideways motion of the foot provides the most discrimination, and a different segment of the gait cycle often leads to a different level of discrimination.

Beside the above work, many other gait-recognition methods have been developed since 2007. In [32], Liu *et al.* employed the dynamic time warping (DTW) as a tool for gait-curve matching. This work was improved in [33], where the wavelet denoising and gait-cycle segmentation algorithms were introduced for data preprocessing. In [34], Trivino *et al.* proposed an approach using a fuzzy finite state machine (FFSM) to model the perception of the signal evolution, which achieved superior results to that of Gafurov and Liu. In [11], Zhang *et al.* proposed a novel algorithm that avoided cycle detection failures and inter-cycle phase misalignment. In [35], Derawi *et al.* provided a stable cycle detection mechanism and improved the gait-based authentication. In [36], an overview of the inertia-based gait recognition methods was given with extensive comparisons.

In [10], [37], [38], a research team from Osaka University conducted extensive research on gait recognition, including the video-based and sensor-based methods. In their sensor-based approach, up to date they have provided the largest inertial sensor-based gait dataset in the world, including 744

subjects (389 males and 355 females) with ages ranging from 2 to 78 years [39], which is a significant contribution to the community of gait research [40]. In their research, the acceleration was found to be better than angular velocity for gait recognition, and the case with distance-normalized obtained significantly better results than that with distance-unnormalized [10].

In recent years, due to the rapid development of mobile devices, the accelerometer and gyroscope have been commonly integrated into the smartphones and smartwatches [41]. It has been possible to use smartphone for gait recognition [9], [15] in a wide range of scenarios, e.g, person authentication [13], [14], [42], medical analysis [43]–[45]. In [13], data were collected on 36 subjects by putting smartphone in the pant's front pocket. The activities of walk, jog, climb-up stairs, and climb-down stairs were included for identification and authentication. In [14], a weighted voting scheme dependent upon the gait characteristic was proposed. The gait characteristic was modeled on the gait frequency, symmetry coefficient, dynamic range and similarity coefficient, based on the accelerometer and gyroscope data. In [42], accelerometer of smartphone was used to capture gait data, and dynamic time warping was employed for gait curve matching. In [44], smartphones were used as health monitors, in which the body motion was predicted based on eight parameters of the phone motion in a gait model. In [46], other than the traditional device-centric and world coordinate systems, a user-centric coordinate system was proposed to represent gait data and better results were obtained in gait recognition.

B. Deep Learning for Gait Recognition

Due to the property of non-replaceability and unique, gait has a wide range of applications in authentication and access control [12], [47], [48]. Meanwhile in recent years, deep learning has made great progress in the field of human gait activities recognition [49], [50]. Unlike traditional machine learning methods, e.g., PCA, MCA [51], SVM [47], [52], [53], etc., deep learning methods perform gait behavior features extraction in a supervised and automatic way and can significantly improve the accuracy of recognition.

We can use a simple deep neural network to extract the motion characteristics of data which is collected from inertial sensors or images, or we can also combine DNN with other traditional machine learning methods [54], [55]. Deep CNN is a kind of deep network that often consists of several convolutional layers, ReLU layers, pooling layers and full connection layers. CNN can extract the abstract features of images and has achieved great success in image recognition processing [56]. Most of the data input for gait recognition is two-dimensional or multi-dimensional data such as images and motion signals.

Due to the outstanding ability of CNN in image processing, many researchers used CNN for gait or activity recognition [49], [57]–[61]. In [49], CNN-based gait recognition was performed by constructing three deep convolutional networks, using the users' gait energy images as input. The gait features were extracted from the bottom, top, and global of the network, which greatly improves the accuracy of the classification. In [62], cross-view gait recognition was studied using CNNs constructed with contrastive loss and triplet ranking loss, which achieved high performance in person verification and identification. In [60], gait data were firstly extracted by a periodogram-based gait separation algorithm, and gait classification and authentication algorithms were then built on convolutional neural networks.

CNN can also be combined with traditional machine learning methods such as PCA, MCA [63], Bayesian classifier [64] and SVM [65]. In [65], CNN was used as a feature extractor, then the extracted features were classified by SVM. At the same time, there are also some researches using CNN to extract three-dimensional data consisting of images and optical flow information for gait and activity recognition [65], [66].

Meanwhile, researchers find that human activities have obvious temporality which contains useful feature for identification, whereas CNN that only processes single time-stamp data tends to ignore it. Therefore, [50] used the 3D data formed from the time-series of 2D images as the input to the CNN, and used 3D convolution kernels to extract feature for activity recognition. In this way, temporal characteristics were also taken into account when extracting the spatial features. It is more common practice to introduce RNN, which records the temporal information of the sequential signal by passing the previous hidden layer state to the current hidden layer. LSTM [67] network is a variation of the RNN. The hidden layer is specially designed and can extract the features of the time-series more effective. [17] proposed to combine LSTM and CNN for activity identification. In this way, it is common to use CNN as a feature extractor, and then use LSTM to further process the gestural features extracted by the CNN [68], [69]. Many application scenarios and comparisons of DNN, CNN, and RNN in the field of gait recognition have been introduced in [70].

In addition to the commonly used CNN and LSTM network models, in order to solve the effects of perspective, weight, clothing and other issues on the extraction of gait intrinsic features, [71] designed a gait feature extractor based on the generative adversarial nets. Sometimes, gait datasets may contain many subjects while each of these subjects only has a small amount of data, which may not support the training of deep models. [72] tried to solve this problem using Siamese neural networks.

The existing researches and results have shown the effectiveness of deep learning in gait and behavior recognition. However, the data of these works are mostly collected under limited and excellent road conditions with specified walking speed. Identification of gait in the unconstrained condition or living environment remains challenging and the ability to combine both CNN and LSTM to extract spatial and temporal information is not powerful enough.

III. Gait Data Collection in the Wild

In this section, we introduce how to collect gait data using smartphones in the wild. First, we will introduce the inertial sensors in the smartphones, then describe the algorithm that partitions the inertial data into walking and non-walking sessions, and finally elaborate the segmentation of gait cycles on the inertial time-series.

A. Inertial Sensors in the Smartphones

Accelerometer and gyroscope are the typical inertial sensors that have been equipped by most smartphones nowa-

days. Either an accelerometer or a gyroscope measures the inertial dynamics in three directions, namely along the X, Y and Z axis. The three-axis accelerometer is based on the basic principle of acceleration, and is used to measure the smartphone's acceleration (including the gravity) in X, Y and Z. The accelerations in the three directions reflect the change of smartphone's linear velocity in the 3D space, and hence reflect the movement of the smartphone users. The three-axis gyroscope captures the angular velocity of a smartphone during its rotation in the space, which can also describe the movement pattern of a user. The smartphones used in our work include Samsung, Xiaomi and Huawei, all of which are installed with the Android operating system. The smartphone itself provides hardware synchronization for accelerometer and gyroscope data. When an user is walking, the smartphone generates accelerations and rotates around different directions applying with the movements of the user. These data are assumed to be individually different, so we collect them as the source data of gait dynamics.

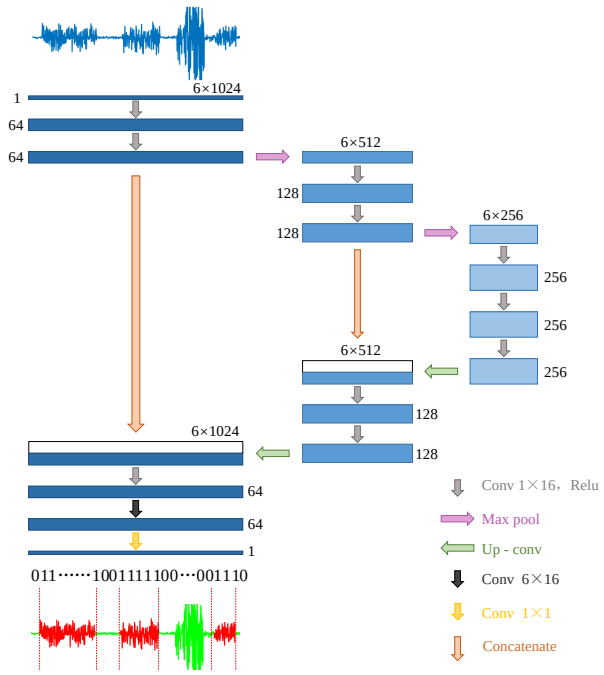


Fig. 1. Architecture of the network for gait data extraction.

B. Gait Data Extraction

The accelerometer and the gyroscope of smartphones are used to collect the inertial gait data. However, without any constrains on the users, we do not know when, where and how the smartphone is used. Consequently, the captured data would consist of the walking session and the non-walking session. While only the walking data are desired for gait feature extraction and person identification, the continuous inertial sequence collected by smartphones in the wild should be partitioned.

Considering that the walking data and non-walking data are semantically different, and the inertial time-series are continuous in both the space domain and time domain, we model the partitioning problem as a time-series segmentation problem. Inspired by U-Net [73], we build a semantic segmentation algorithm with one-dimensional deep convolutional neural networks. The architecture of the proposed network is shown in Fig. 1, and the details are shown in Table I. In order to improve the accuracy of segmentation, we fuse hierarchical convolutional features at multiple stage together in the network.

TABLE I
DETAILS OF THE DATA EXTRACTION NETWORK STRUCTURE

Layer Name	Kernel Size	Kernel Num.	Stride	Feature Map
conv1_1	1×16	64	1	6×1024×64
conv1_2	1×16	64	1	6×1024×64
pool1	1×2	/	2	6×512×64
conv2_1	1×16	128	1	6×512×128
conv2_2	1×16	128	1	6×512×128
pool2	1×2	/	2	6×256×128
conv3_1	1×16	256	1	6×256×256
conv3_2	1×16	256	1	6×256×256
conv3_3	1×16	256	1	6×256×256
upconv1	1×2	128	1	6×512×128
concat1	/	/	/	6×512×256
conv4_1	1×16	128	1	6×512×128
conv4_2	1×16	128	1	6×512×128
upconv2	1×2	64	1	6×1024×64
concat2	/	/	/	6×1024×128
conv5_1	1×16	64	1	6×1024×64
conv5_2	6×16	64	1	1×1024×64
conv5_3	1×1	1	1	1×1024×1

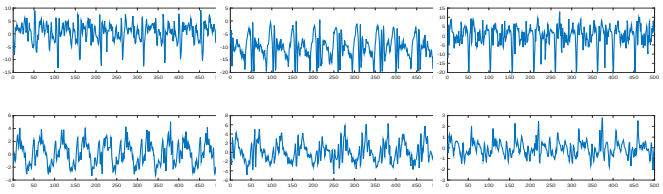
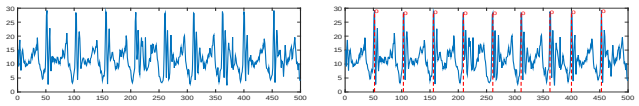


Fig. 2. The sensors data, the top line is ACC_x, ACC_y, and ACC_z from left to right, and the bottom line is GYR_x, GYR_y, and GYR_z from left to right.



(a) (b)

Fig. 3. An example of step segmentation. (a) is the gait curve ACC_o, based on the three components as shown in Fig. 2. (b) shows the segmentation results, where the red dots denote the local maximums and divide the steps.

C. Gait Cycle Segmentation

As has been observed by many previous work, cycle-partitioned gait data can often generate improved performance over non-cycle-partitioned ones [7]. We also validate this point in our experiments, which will be introduce in section V. Therefore, it is necessary to segment the extracted gait data by walking steps.

Once gait data are extracted by the proposed deep semantic segmentation network, we divide the continuous ones into separate steps. Note that, in this paper, a complete step refers to touching the ground with the same foot twice in succession. A sample gait data is shown in Fig. 2, which includes three accelerometer curves and three gyroscope curves. Without loss of generality, we select accelerometer data as a basis for the partitioning task. As the smartphone can be placed in random directions by the user, one single axis of the acceleration or gyroscope cannot

stably reflect the fluctuations of the gait curve. Meanwhile, the absolute acceleration value of the axis perpendicular to the ground is the largest among the three values. In order to remove the influence of the phone's posture, we process the triaxial acceleration data to get ACC_o as the basis for gait cycle segmentation. The ACC_o is calculated by $ACC_o = \sqrt{ACC_x^2 + ACC_y^2 + ACC_z^2}$, where ACC_x, ACC_y and ACC_z denote the data values of the acceleration value in x, y, and z, respectively. Fig. 3(a) shows the ACC_o calculated based on ACC_x, ACC_y and ACC_z in Fig. 2. Finally, we find the step-separation points on the local maximums on the ACC_o curve. We set the threshold for each subject's gait cycle (period length and amplitude extremum) to automatically extract the gait cycle. Specifically, based on our analysis of human steps, the step-separation points are recognized by the following rules:

- The point is a local maximum on the ACC_o curve, at a range of 0.8s.
- The ACC_o value is larger than $10m/s^2$. It is because that the acceleration of a falling movement when a person walks should be at least greater than the gravity acceleration.
- The time gap between two consecutive local maxima should be between 0.8s and 1.6s. It is because that, it takes about 0.8s to 1.6s for a normal person to complete one single step, which is an empirical value obtained by manually observing a small part of the data from each subject in 118 subjects.

IV. Gait Recognition with Deep Neural Networks

In biometrics, gait recognition has meanings of two-fold. One is gait identification, which identifies the identity of a sample within a given number of candidate identities. The other is gait authentication, which judges if two samples belong to the sample identity or not. Usually, the former can be modeled as an n-class classification problem, while the latter is often modeled as a binary-classification problem. As have been discussed in the Section I, we investigate deep learning based techniques for solving these two problems. To be specific, we first present deep neural networks for gait identification, and then for gait authentication in this section.

At present, there are mainly two types of deep neural networks that have powerful performance and wide application. One is the deep convolutional neural network (DCNN), and the other is the deep recurrent neural network (DRNN). DCNNs are built on the convolution and pooling operations, often in an alterative way. The feature maps will become smaller but more concise, which makes the DCNNs be very effective in feature abstraction from arrayed signals such as an image. DRNNs process the input signal in a recursive way, in which the information at a later time-stamp can be related to the information at an earlier time-stamp for prediction. This characteristic makes the DRNNs very talent in handling time-series signals such as voice and speech. A representative DRNN model is the LSTM (Long Short-Term Memory), which has a memory gate and a forget gate to control the extent of information interaction.

For the tri-axis accelerometer gait data and gyroscope gait data, they can be arranged together into a six-axis inertial gait data. Then, the six-axis gait data are featured both as arrayed signals and as time-series signals. As a results, both DCNN and DRNN can process them effectively. In order to fully use the strengths of DCNN and DRNN, we purposely employ the CNN and LSTM for effective learning of the spatial feature and temporal feature, respectively, and perform information fusion to improve the gait feature representation and gait classification.

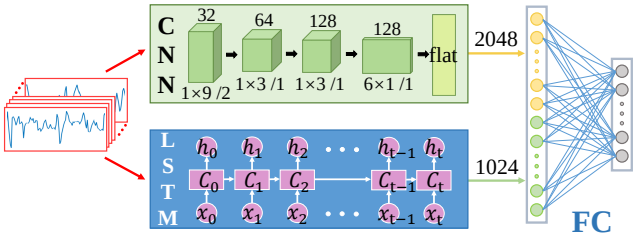


Fig. 4. The proposed network architecture for gait identification.

Specifically, in the proposed architecture, the fusion of DCNN and DRNN is performed in a feature-concatenation way. The inertial data are processed by a DCNN and an LSTM, independently. The extracted features are concatenated, and linked to a successive layer in a full-connection manner for gait classification. Fig. 4 shows the proposed network architecture.

A. Neural Network for Identification

*1) **Problem formulation**:* Given an inertial gait curve $\mathbf{x}$ with a sampling length of T, then $\mathbf{x}$ can be expressed as

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, ..., x_T), \\ \text{w.r.t.}, x_t &= (acc_x^t, acc_y^t, acc_z^t, gyr_x^t, gyr_y^t, gyr_z^t)^\top, \end{aligned} \tag{1}$$

where $(acc_x^t,\ acc_y^t,\ acc_z^t)$ and $(gyr_x^t,\ gyr_y^t,\ gyr_z^t)$ denote the accelerometer and gyroscope components in the X, Y and Z axis at time-stamp t, respectively. Then the problem is how to recognize the identity of the subject based on the input data $\mathbf{x}$. To formulate this problem, let's suppose $\mathbf{s}$=$(s_1, s_2, ..., s_n)$ be a number of n candidate subjects, s_i be the ith subject, then the output can be represented as an n-dimension vector,

$$\mathcal{O} = (o_1, o_2, ..., o_n), \tag{2}$$

where o_i=$P(s_i|\mathbf{x})$, i.e., the possibility that $\mathbf{x}$ belongs to s_i. Let's further suppose s be the identity of the input data $\mathbf{x}$, then it is formulated as

$$s = arg\max_{s_i}\{o_i \mid 1 \le i \le n\}. \tag{3}$$

As a result, to solve the problem of gait identification, we have to associate the maximum possibility values to the corresponding subjects.

*2) **Network structure**:* As illustrated by Fig. 4, the gait-identification network consists of a CNN and an LSTM, which are parallel to each other. The CNN and LSTM work as two feature extractors which get the corresponding features – $feat_{cnn}$ and $feat_{lstm}$. The CNN and LSTM are followed by a full connection layer, which works as a classifier and uses the feature vector concatenated by $feat_{cnn}$ and $feat_{lstm}$ as the input.

LSTM network. LSTM is an improvement to RNN and still has the basic structure of RNN. For an RNN network with L hidden layers, given a gait sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$, a state $\mathbf{h}_t^l$ will be generated for each layer at time t:

$$\mathbf{h}_t^l = \sigma(W_{xh}^l x_t + \mathbf{h}_{t-1}^l W_{hh}^{tl} + \mathbf{h}_t^{l-1} W_{hh}^{ll} + \mathbf{b}_h^l), \quad (4)$$

where $\mathbf{h}_t^l$ is the state of layer l at time t, x_t is input at time t, W_{xh}^l is the weight matrix of the input x_t to the lth hidden layer, W_{hh}^{tl} is the weight matrix of state at time t-1 to state at time t at the same layer l, W_{hh}^{ll} is the weight matrix of state at layer l-1 to state at layer l at the same time t, $\mathbf{b}_h^l$ is the bias of layer l, and $\sigma(\cdot)$ is the activation function.

For an LSTM network, the basic unit is composed of a cell, an input gate, an output gate and a forget gate. Similar to RNN, we can also examine the LSTM network with a state $\mathbf{h}_t^l$. In order to obtain a better memory effect for information interaction along the time-series, an input gate $\mathbf{i}$, an forgetting gate $\mathbf{f}$, a state vector $\mathbf{c}$, and an output gate $\mathbf{o}$ are added to the hidden layer state. Then, the $\mathbf{h}_t^l$ can be updated as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma_i(W_{xi}x_t + W_{hi}^t \mathbf{h}_{t-1}^l + W_{hi}^l \mathbf{h}_t^{l-1} + W_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i),\\
\mathbf{f}_t &= \sigma_f(W_{xf}x_t + W_{hf}^t \mathbf{h}_{t-1}^l + W_{hf}^l \mathbf{h}_t^{l-1} + W_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f),\\
\mathbf{c}_t &= \mathbf{f}_t + \mathbf{c}_{t-1}\sigma_i(W_{xc}x_t + W_{hc}^t \mathbf{h}_{t-1}^l + W_{hc}^l \mathbf{h}_t^{l-1} + \mathbf{b}_c),\\
\mathbf{o}_t &= \sigma_o(W_{xo}x_t + W_{ho}^t \mathbf{h}_{t-1}^l + W_{ho}^l \mathbf{h}_t^{l-1} + W_{co}\mathbf{c}_{t-1} + \mathbf{b}_o),\\
\mathbf{h}_t^l &= \mathbf{o}_t\sigma_h(\mathbf{c}_t),
\end{aligned} \quad (5)$$

where all the W, σ, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$ are the parameter of layer l, as all of them have a hidden superscript 'l'. W_{xi} is the weight matrix of the input x_t to the input gate. σ_i is the activation function of the input gate. $\mathbf{b}_i$ is the bias of the input gate. The meaning of other W, $\mathbf{b}$, and σ can be inferred from the above rule. Given the LSTM network is constructed with L hidden layers, with each containing N hidden nodes, then, for each input $\mathbf{x} = (x_1, x_2, ..., x_T)$, the output feature can be formulated as

$$\begin{aligned}
feat_{lstm} &= \mathbf{h}_T^L \\
&= (f_1, f_2, ..., f_N).
\end{aligned} \quad (6)$$

CNN network. Considering that the input signals are the time-series, a number of one-dimension kernels are used in the convolution operations in the proposed CNN network. Specifically, the proposed CNN network is constructed with 4 convolution layers and 2 max-pooling layers. The convolution kernel abstracts the feature of gait curve along the time-series and the max-pooling downsamples the feature map in the pooling window. Table II describes the detailed structure of the proposed CNN network. Following the CNN structure, the output feature has a dimension of 1×16×128. Then, we flatten the output convolutional feature map to a one-dimension feature vector $feat_{cnn}$.

Fully Connected Layer. The fully connected layer is the concatenation of the features extracted by LSTM and CNN, i.e., $feat_{full}$=($feat_{lstm}$; $feat_{cnn}$). Then, a softmax operation is applied to produce the classification output, as formulated by Eq. (7),

$$\mathbf{o} = \mathbf{Softmax}(feat_{full} * W_o + \mathbf{b}_o), \quad (7)$$

where W_o is the weight matrix of the output layer, $\mathbf{b}_o$ is the bias of the output layer.

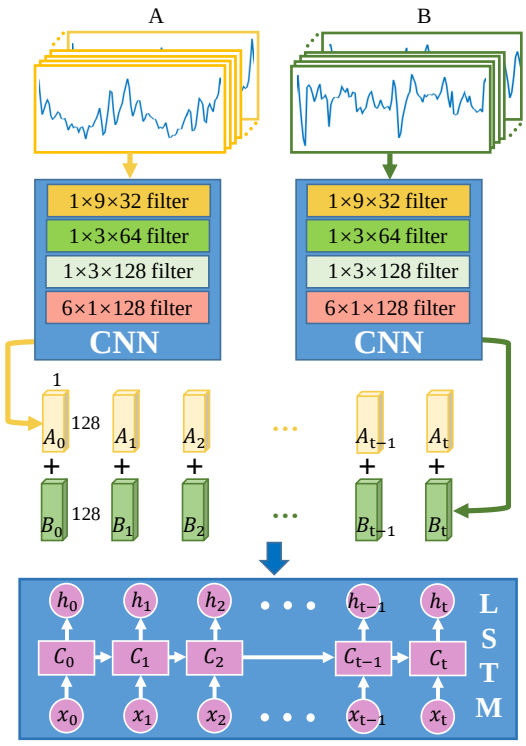


Fig. 5. Network architecture for gait authentication.

TABLE II
DETAILS OF THE CNN STRUCTURE

Layer Name	Kernel Size	Kernel Num.	Stride	Feature Map
conv1_1	1×9	32	2	6×64×32
pool1	1×2	/	2	6×32×32
conv2_1	1×3	64	1	6×32×64
conv2_2	1×3	128	1	6×32×128
pool2	1×2	/	2	6×16×128
conv3_1	6×1	128	1	1×16×128

*3) **Loss Function**:* For each input sample $\mathbf{x}$, the predicted output of the network is $\mathbf{o} = (o_1, o_2, ..., o_n)$, with $o_i = P(s_i|\mathbf{x})$. The value of o_i is between 0 and 1, and the larger the value, the greater the probability that $\mathbf{x}$ belongs to s_i. Based on the output $\mathbf{o}$, we can get the class label as

$$\mathbf{o}' = (o_1', o_2', ..., o_n'), \quad (8)$$

where

$$o_i' = \begin{cases} 1 & \mathbf{x} \in s_i, \\ 0 & other. \end{cases} \quad (9)$$

Then, we can construct the training loss with cross entropy, as formulated by Eq. (10),

$$\mathcal{L}(\mathbf{o}, \mathbf{o}') = \sum_i^n o_i' \ln o_i + (1 - o_i') \ln(1 - o_i). \quad (10)$$

As can be seen from Eq. (10) that, the cross entropy is a positive number. When $o_i \approx 0, o_i' = 0$, or $o_i \approx 1, o_i' = 1$, the cross entropy will be small. In other words, a larger difference between o_i and o_i' will result in a larger cross-entropy value. This property will help the convergence of the network in the training. Meanwhile, using the cross-entropy cost function instead of the variance cost function can speed up the training procession.

B. Neural Network for Authentication

Let two sequences of gait data $\mathbf{x_a}$ and $\mathbf{x_b}$ be the input of the authentication network, which are expressed as

$$\begin{aligned} \mathbf{x_a} &= (x_{a,1}, x_{a,2}, ..., x_{a,T}), \\ \mathbf{x_b} &= (x_{b,1}, x_{b,2}, ..., x_{b,T}), \end{aligned} \tag{11}$$

where

$$x_t = (ACC_x^t, ACC_y^t, ACC_z^t, GYR_x^t, GYR_y^t, GYR_z^t)^\top, \tag{12}$$

and T is the length of the input sequence. As discussed in the beginning of Section IV, the authentication is formulated as a binary-classification problem. The output of the network is set as two dimensions. We use 'True' and 'False' to denote the input data are from the same subject and different subjects, respectively.

In order to fully use the advantage of CNN and RNN, we use the CNN as a feature extractor which maps the input inertial signals into lower-dimension abstractions. In our design, we use the CNN trained on dataset from 98 subjects in the classification in Section IV-A as the feature extractor. These 98 subjects have no overlap with the 20 subjects of test in the authentication. Fig. 5 shows the structure of the authentication network, where the CNN is fixed as a feature extractor. Given the size of the input gait signal is 6×128, then the output of the CNN is $1\times16\times128$. For the consequent LSTM computation, we have to make sure that the input signals have the property of time-series. Therefore, we rearrange the CNN feature into 16×256 features, which are divided into 16 blocks, with each block contains a 256-dimension feature vector. The 16 blocks of features are then fed into a double-layer LSTM for training and prediction. For the CNN network, the weights are fixed as that have been trained in the identification network.

V. Experiments and Results

In this section, we first introduce six datasets that contain inertial gait data collected using smartphones in the wild. Then, we will describe the experimental settings, including the way of data alignment, the selection of comparison methods, and the strategy of training, etc. And finally, we will report the evaluation results for both the identification case and the authentication case.

A. Datasets

Generally, deep-learning methods require a large number of samples for training, and existing datasets cannot meet the demand. Meanwhile, there are few datasets collecting unconstrained inertial data in living environments for gait recognition. We collect the inertial gait data in the wild, where the subjects are not limited to walking on specific roads or speeds. Data are collected in daily life, such as walking after meals. Note that, all the data collected have been pre-processed by the gait-extraction algorithm introduced in Section III-B. A number of 118 subjects are involved in the data collection. Among them, 20 subjects collect a larger amount of data in two days, with each holding thousands of samples, and 98 subjects collect a smaller amount of data in one day, with each having hundreds of samples. Each data sample contains the 3-axis acceleromter data and the 3-axis gyroscope data. The sampling rate of all sensor data is 50Hz. According to the different evaluation purposes, we construct six datasets[1] based on the collected data.

1) Dataset #1: This dataset is collected on 118 subjects. Based on the step-segmentation algorithm introduced in Section III-B, the collected gait data can be annotated into steps. Following the findings that two-step data have a good performance in gait recognition [7], we collected gait samples by dividing the gait curve into two continuous steps. Meanwhile, we interpolate a single sample into a fixed length of 128 using the linear interpolation function. In order to enlarge the scale of the dataset, we make a one-step overlap between two neighboring samples for all subjects. In this way, a total number of 36,884 gait samples are collected. These samples are sorted by time. For each subject, we select the first 90% samples for training, and the rest 10% for test. There are 33,104 training samples and 3,740 test samples, without overlap between the two subsets.

2) Dataset #2: This dataset is collected on 20 subjects. We also divide the gait curve into two-step samples and interpolate them into the same length of 128. As each subject in this dataset has a much larger amount of data as compared to the that in Dataset #1, we do not make overlap between the samples. Finally, a total number of of 49,275 samples are collected, in which 44,339 samples are used for training, and the rest 4,936 for test.

3) Dataset #3: This dataset is collected on the same 118 subjects as in Dataset #1. Different from Dataset #1, we divide the gait curve by using a fixed time length, instead of a step length. Exactly, we collect a sample with a time interval of 2.56 seconds. While the frequency of data collection is 50Hz, the length of each sample is also 128. Also, we make an overlap of 1.28 seconds to enlarge the dataset. A total number of 29,274 samples are collected, in which 26,283 samples are used for training, and the rest 2,991 for test.

4) Dataset #4: This dataset is collected on 20 subjects. We also divide the gait curve in an interval of 2.56 seconds. We make no overlap between the samples. Finally, a total number of of 39,314 samples are collected, in which 35,373 samples are used for training, and the rest 3,941 for test.

5) Dataset #5: This dataset is used for authentication. It contains 74,142 authentication samples of 118 subjects, where the training set is constructed on 98 subjects and the test set is constructed on the other 20 subjects. There are 66,542 samples and 7,600 samples for training and test, respectively. Each authentication sample contains a pair of data sample that are from two different subjects or one same subject. The data sample consists of a 2-step acceleration and gyroscopic data, which are interpolated in the way as described in Dataset #1 and Dataset #2. The two data samples are horizontally aligned to create an authentication sample.

6) Dataset #6: This dataset is also used for authentication. The authentication samples are constructed as the same as in Dataset #5. The only difference is that, in authentication sample construction, two data samples from two subjects are vertically aligned instead of horizontally aligned.

Table III shows the detail information of these six datasets.

[1] Available at https://sites.google.com/site/qinzoucn

TABLE III
DETAIL INFORMATION OF THE SIX DATASETS.

Dataset Name	Usage	Number of Subjects	Time-fixed or Interpolation	Overlap in Sampling	Samples for Training	Samples for Test	Alignment
Dataset #1	Classification	118	Interpolation	1 step	33,104	3,740	N/A
Dataset #2	Classification	20	Interpolation	0	44,339	4,936	N/A
Dataset #3	Classification	118	Time-fixed	1 step	26,283	2,991	N/A
Dataset #4	Classification	20	Time-fixed	0	35,373	3,941	N/A
Dataset #5	Authentication	118	Interpolation	1 step	66,542	7,600	Horizontal
Dataset #6	Authentication	118	Interpolation	1 step	66,542	7,600	Vertical

*Note: there is no overlap between the training sample and the test sample for all datasets.

B. Implementation Details

1) Data collection: We develop an APP and installed it on the smartphone with Android platform. Several brands of smartphones have been used in our experiment, including the Samsung, Xiaomi and Huawei. The frequency of the accelerometer and the gyroscope is set to 50Hz. At the same time, it records the data of these two sensors in real time. When using the APP, the user inputs his own identity information and start the data collection process. The users can put the phone in his/her trouser pocket, play it in hand, or place it on the desk, without any constrains. It is worth noting that the APP have to collect data for a long period of time, making sure that the captured data contain enough walking data. Meanwhile, the APP will automatically get rid of the data if the smartphone is static within a period of 3 seconds. The captured data have seven dimensions, including the time stamp, the triaxial values of the acceleration sensor and the triaxial values of the gyroscope.

2) Experimental Settings: In gait classification, a number of network structures have been designed, including the LSTM-based, the CNN-based, and the 'CNN+LSTM' based, and their performances have been compared under an evaluation metric of accuracy.

For LSTM-based methods, each hidden layer in the LSTM has a number of N=64 hidden nodes. The learning rate is set to 0.0025, and the number of epoches for training is 200.

For CNN-based methods, the six-axis interpolation data are used as the input, with the data shape of 6×128. The classification experiments are conducted on the first four datasets as introduced in Section V-A. In training the CNN, the learning rate is 0.0025, and the number of epoches for training is 200.

3) Authentication experiment: As has been introduced in Fig. 5, the authentication network contains a CNN and an LSTM. In the training process, parameters of CNN are frozen, and the LSTM network equipped with 64-node hidden layers is trained with a learning rate of 0.0025, an epoch number of 300 and a batch size of 1,500.

C. Performance on Gait Data Extraction

1) Datasets: Two datasets are constructed for evaluation of the proposed gait-data-extraction method. Basic information of the two datasets have been shown in Table IV, and the details are given as below:

- Dataset #7: it contains 577 samples of 10 subjects, with data shape of 6×1024. Among these samples, 519 are used for training and 58 are for test. Both the training and test samples are from the 10 subjects.
- Dataset #8: it contains 1,354 samples of 118 subjects, with data shape of 6×1024. Among these data, 1022 samples from 20 subjects are used for training, and 332 samples from the other 98 subjects are used for test.

TABLE IV
DETAIL INFORMATION OF THE GAIT-DATA EXTRACTION DATASETS

Dataset Name	Number of Subjects	Samples for Training	Samples for Test
Dataset #7	10	519	58
Dataset #8	118	1022	332

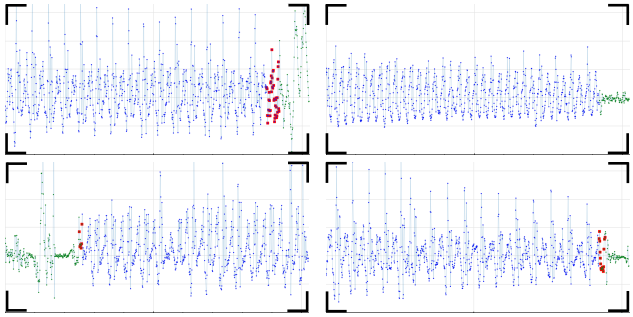


Fig. 6. Four examples of walking data extraction using the proposed method. Note that, the blue points denote the walking data, green points denote the non-walking data, and the red denotes the false classified.

For both datasets, each sample is attached with a label file, which contain 1024 binary values, with 1 as the walking data, and 0 as the non-walking data. The labels are manually annotated.

2) Experimental results: We train the CNN network proposed in Section III-B for gait data extraction. For Dataset #7 and Dataset #8, the learning rate is set as 0.0001, and the number of training epoches is set as 150. Fig. 6 shows four sample results obtained by the proposed network. It can be seen from Fig. 6, most data are correctly classified, a small portion of some walking data are extracted as non-walking (red on blue), and also a small portion of non-walking data are extracted as walking (red on green). The misclassification occurs at the transition area between the walking section and the non-walking section. It is reasonable since there are uncertainties for those points at the transition area.

Specifically, on Dataset #7, where the training data and test data have no overlap but are all from the same 10 subjects, the proposed method achieves an accuracy of

90.22%. It shows the effectiveness of the proposed method in separating walking data from the non-walking data. On Dataset #8, where the training data and the test data are from different subjects, an accuracy of 85.57% is obtained. It indicates that the proposed method has a high generalization power.

D. Performance of LSTMs at Different Data Settings

In this experiment, we examine how the network structures influence the performance of LSTMs, and how effective the different data settings are for classification.

1) Different LSTM networks: The layers of LSTM can be constructed with information propagation in forward direction only or in both forward and backward directions. In this experiments, we test three LSTM network architectures, that are:

- SL-LSTM: an LSTM with one single hidden layer.
- Bi-LSTM: a bi-directional LSTM, with a layer forward and a layer backward.
- DL-LSTM: an LSTM with two hidden layers.

2) Different data settings: The original data contain six channels. We will investigate how the combination of channel(s) will affect the performance. These data will be constructed based on Dataset #1, Dataset #2, Dataset #3 and Dataset #4. Specifically, we build three network structures, with each evaluated on 8 different combinations of data channels, i.e., 4 for the interpolated data and 4 for the time-fixed data. The four interpolated data are:

- interp_6: three-axis accelerometer data and three-axis gyroscope data of Dataset #1 and Dataset #2, as have been sampled in a interpolation way, with data shape of 6*128.
- interp_acc: three-axis accelerometer data of Dataset #1 and Dataset #2, with data shape of 3×128.
- interp_gyr: three-axis gyroscope data of Dataset #1 and Dataset #2, with data shape of 3×128.
- interp_sqrt: the mean square root of the three-axis accelerometer data of Dataset #1 and Dataset #2, with data shape of 1×128.

And the four time-fixed data are:

- fixed_6: three-axis accelerometer data and three-axis gyroscope data of Dataset #3 and Dataset #4, as have been sampled in a time-fixed way, with data shape of 6×128.
- fixed_acc: three-axis accelerometer data of Dataset #3 and Dataset #4, with data shape of 3×128.
- fixed_gyr: three-axis gyroscope data of Dataset #3 and Dataset #4, with data shape of 3×128.
- fixed_sqrt: the mean square root of the three-axis accelerometer data of Dataset #3 and Dataset #4, with data shape of 1×128.

3) Experimental details and results: For the above LSTM networks, the number of nodes for the hidden layer is set to 64. The last hidden layer is followed by a fully connected layer, which is used as a classification output layer. The size of the fully connected layer is 64×20 for the case of 20 subjects or 64×118 for the case of 118 subjects. All networks are trained with a learning rate of 0.0025, and a training epoch of 300.

Fig. 7 shows the results of the three LSTM networks at eight different data settings. The data are all constructed based on the Dataset #1 and Dataset #3, which have 118 subjects. It can be found from Fig. 7 that, the results on accelerometer data are better than that on gyroscope data, and the results on gyroscope data are better than that on the mean square root data. It simply indicates that, the accelerometer sensor can better capture the gait feature than the gyroscope sensor.

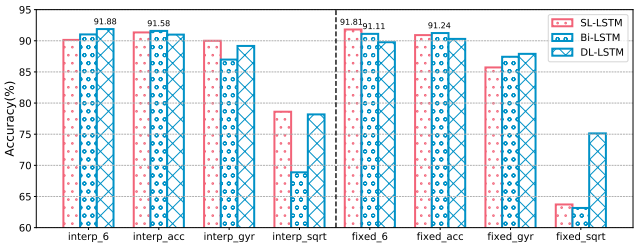


Fig. 7. Performance of different LSTM networks. The classification experiments are conducted on 118 subjects. For each group of results, the left, middle, and right bars correspond to the results of the single-layer LSTM (SL-LSTM), the bi-directional LSTM (Bi-LSTM) and the double-layer LSTM (DL-LSTM), respectively.

From Fig. 7 we can also observe that, results on interpolated data are slightly better than that on time-fixed at the according settings. The best results on both the interpolated data and the time-fixed data are obtained on the six-axis cases by the double-layer LSTM – DL-LSTM. It indicates that, the accelerometer data and the gyroscope data can be complementary to each other to better represent the gait features. While DL-LSTM gets the highest classification accuracy of 91.88% at the interp_6, we will use DL-LSTM and the interp_6 data in our later experiments for comparison.

E. User-Identification Performance

In this subsection, we introduce several experiments that evaluate the performance of various methods in gait identification. As a classification problem, we use the accuracy as a metric in performance evaluation. The accuracy is defined as,

$$\textbf{Accuracy} = \frac{\textbf{Correctly Classified Samples}}{\textbf{Total Testing Samples}}. \quad (13)$$

1) Comparison methods: To evaluate the proposed methods in gait-based person identification, a number of ten methods are included for comparison, that are:

- Fourier [74]: the gait data is first processed by an auto-correlation operation, and then the results are converted into the frequency domain using FFT. In identification, the first 40 FFT coefficients per channel are selected as the gait features. A one-vs-all SVM is employed for classification.
- Wavelet [9]: the gait data are decomposed by using the Mexican Hat Wavelet, and the low-frequency part of the results are used as gait feature for person identification. A one-vs-all SVM classifier is employed.
- EigenGait [7]: the inertial gait data are decomposed in the eigen space, and the principle components are taken as gait features for person identification [7]. A one-vs-all SVM classifier is employed.

TABLE V
CLASSIFICATION PERFORMANCE OF TRADITIONAL METHODS

Dataset	Channels	EigenGait	Wavelet	Fourier
Dataset #2 (20 subjects)	interp_6	87.03%	87.20%	93.64%
	interp_acc	86.95%	88.84%	90.15%
	interp_sqrt	65.24%	78.40%	66.90%
Dataset #1 (118 subjects)	interp_6	76.58%	75.13%	81.55%
	interp_acc	76.66%	78.96%	75.24%
	interp_sqrt	37.62%	52.25%	35.72%

- LSTM: it is the DL-LSTM introduced in Section V-D, and has to be trained from scratch.
- CNN: it is the convolutional neural network introduced in Section IV-A, and has to be trained from scratch.
- CNN+LSTM: it is the network introduced in Fig. 4, which combines the above two networks. The whole network has to be trained from scratch.
- CNN$_{fix}$+LSTM: it is also the network introduced in Fig. 4. When training, the parameters of CNN are fixed as that in the CNN model that has been trained independently, and the parameters of the LSTM and fully connected layer have to be trained from scratch.
- CNN+LSTM$_{fix}$: it is also the network introduced in Fig. 4. When training, the parameters of LSTM are fixed as that in the LSTM model that has been trained independently, and the CNN and fully connection layer have to be trained from scratch.
- IdNet [63]: a CNN-based person-identification method using inertial data collected by smartphone. The IdNet method is trained from scratch. The dataset, named as IdNet dataset, is constructed on 50 subjects, with 15,096 samples for training and 6,471 samples for test.
- DeepConvLSTM [68]: a deep-learning framework composed of convolutional and LSTM recurrent layers. It is capable of automatically learning feature representations and modelling the temporal dependencies between their activation. It is trained from scratch.

The Fourier, Wavelet and EigenGait methods were selected for comparison since they are commonly used for time-series signal analysis, and are often used for feature extraction in the study of gait recognition. The IdNet and DeepConvLSTM were selected for comparison. It is because these two deep learning based methods represent the state-of-the-art gait recognition methods, and can make the comparison more extensive.

2) Performance of traditional methods: The classification results of three traditional methods, i.e., Fourier, Wavelet and EigenGait, have been shown in Table V. It can be observed that, the Fourier-transform-based method gets the best performance among the three methods, on both Dataset #1 and Dataset #2, where the accuracy values are 81.55% on Dataset #1 and 93.64% on Dataset #2. These best performances are obtained on the interpolation data with six-axis inertial inputs.

3) Performance of deep learning methods: Three datasets, i.e., Dataset #1, Dataset #2 and the IdNet dataset, are used to evaluate the seven deep learning based methods: LSTM, CNN, 'CNN+LSTM', 'CNN$_{fix}$+LSTM', 'CNN+LSTM$_{fix}$', IdNet and DeepConvLSTM. The classification results are shown in Table VI. It can be seen from Table VI that, all the seven deep-learning methods achieve over 91.8% accuracy on Dataset #1 (with 118 subjects), and over 96.7% accuracy on Dataset #2 (with 20 subjects), which are much higher than that obtained by the traditional methods. While on the IdNet dataset, all the seven methods achieve over 99.2% accuracy, and they have very little difference on the classification results. This is because the data in IdNet dataset are collected in relatively standard walking style, with less classification challenge.

It can also be observed from Table VI, the CNN obtains higher performance than the LSTM, on both Dataset #1 and Dataset #2. For the more challenging 118-subject case, the CNN outperform LSTM by about 1% in accuracy. It indicates that, the CNN can better extract the gait features from the inertial gait curves. The results show that the 'CNN+LSTM' trained from scratch will not guarantee an improve performance than CNN or LSTM. One possible reason may be that, the CNN and LSTM are parallel in the identification network, and one final loss may not reflect the real status of one single network. That is to say, in the training process, the gradient back-propagation may be right for one network while be wrong for the other.

When fixing the weight parameters of one network and training the other, we get improved performance over single network based methods on Dataset #1 and Dataset #2. This is because, the final loss can directly reflect the status of the unfixed network, and the two types of features are complementary to each other in the classification. We can also see that, 'CNN+LSTM$_{fix}$' achieves the highest performance among the methods, and outperforms the 'CNN$_{fix}$+LSTM' by about 0.6% and 0.3% on Dataset #1 and Dataset #2, respectively. This may be because the LSTM is much more difficult to train than the CNN in a parallel-structured network. From the results we can see, IdNet obtains an accuracy about 0.6% and 0.5% lower than 'CNN+LSTM$_{fix}$' on Dataset#1 and Dataset#2, respectively. While for DeepConvLSTM, the accuracy is about 1.2% and 0.5% lower than 'CNN+LSTM$_{fix}$'.

F. User-Authentication Performance

1) Comparison methods: A number of eleven methods are included for comparison in the authentication experiments, that are:

- Fourier [74]: the first 80 FFT coefficients per channel are selected in Dataset #5 as the gait features, while in Dataset #4 it is 40 per channel. A 2-class SVM is employed as a classifier.
- Wavelet [9]: the continuous wavelet transform is employed to obtain the components from scale 1 to 20. The energy of the frequency band signal at scale i is $E_i = (\sum_{k=1}^{M} |x_i(k)|^2)^{\frac{1}{2}}$, where $x_i(k)$ denotes the discrete point amplitude of the reconstructed signal at scale i, and M denotes the number of discrete points. In our case, M=256 is for the horizontally spliced, and M=128 is for the vertically spliced. We construct the feature vector as

$$\mathbf{F} = [\frac{E_1}{E}, \frac{E_2}{E}, ..., \frac{E_{20}}{E}], \tag{14}$$

with E=$(\sum_{i=1}^{20} {E_i}^2)^{\frac{1}{2}}$, which places an L2 normalization. A 2-class SVM classifier is employed.

TABLE VI
CLASSIFICATION PERFORMANCE OF DEEP-LEARNING METHODS

Classification Methods	Dataset #1 (118 Subjects)	Dataset #2 (20 Subjects)	IdNet Dataset (50 Subjects)
IdNet [63]	92.91%	96.78%	99.58%
DeepConvLSTM [68]	92.25%	96.80%	99.24%
LSTM	91.88%	96.98%	99.46%
CNN	92.89%	97.02%	99.71%
CNN+LSTM	92.51%	96.82%	99.61%
CNN$_{fix}$+LSTM	92.94%	97.04%	99.64%
CNN+LSTM$_{fix}$	93.52%	97.33%	99.75%

- EigenGait [7]: the principle components generated by the eigen decomposition are taken as gait features for person authentication. A 2-class SVM classifier is employed.
- CNN_horizontal: the CNN network using horizontally aligned data pairs as the input.
- CNN_vertical: the CNN network using vertically aligned data pairs as the input.
- LSTM_horizontal:the LSTM network using horizontally aligned data pairs as the input.
- LSTM_vertical: the LSTM network using vertically aligned data pairs as the input.
- CNN+LSTM_horizontal: the 'CNN+LSTM' network, as have been introduced in Fig. 5, using horizontally aligned data pairs as the input. The weight parameters of CNN are unfixed in the training.
- CNN+LSTM_vertical: the 'CNN+LSTM' network using vertically aligned data pairs as the input. The weight parameters of CNN are unfixed in the training.
- CNN$_{fix}$+LSTM_horizontal: the 'CNN$_{fix}$+LSTM' network, as have been introduced in Fig. 5, using horizontally aligned data pairs as the input. The weight parameters of CNN are fixed in the training.
- CNN$_{fix}$+LSTM_vertical: the 'CNN$_{fix}$+LSTM' network using vertically aligned data pairs as the input. The weight parameters of CNN are fixed in the training.

Note that, CNN$_{fix}$ is pre-trained with data samples of 98 subjects in Dataset #1. These 98 subjects are the ones used for training in Dataset #5 and Dataset #6. As a result, the subjects of the test samples in Dataset #5 and Dataset #6 are unseen to the CNN$_{fix}$, which makes the authentication-task very challenging.

2) Metric: The accuracy is also employed as a metric to evaluate the performance of various methods. Meanwhile, the ROC curve is also employed for the comparison. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at varying threshold settings. The TPR and FPR are defined as,

$$\mathbf{TPR} = \frac{\mathbf{True\ Positive}}{\mathbf{True\ Positive} + \mathbf{False\ Negative}}, \quad (15)$$

$$\mathbf{FPR} = \frac{\mathbf{False\ Positive}}{\mathbf{False\ Positive} + \mathbf{True\ Negative}}. \quad (16)$$

3) Performance: Note that, for all authentication methods, except the EigenGait[2], the input data can be aligned in two manners. One is in horizontal, and the other is in vertical. For example, with two 6×128 samples, the input data can be 6×256 as aligned in horizontal or 12×128 as aligned in vertical. In the experiments, 66,542 pairs of samples from 98 subjects are used for training, and 7,600 pairs of samples from the other 20 subjects are used for test. Positive and negative samples each account for half.

[2]It can only decompose feature vectors with a pre-defined dimension.

TABLE VII
PERFORMANCE OF AUTHENTICATION (IN ACCURACY)

Authentication Methods	Dataset #5 (Horizontal)	Dataset #6 (Vertical)
CNN	78.47%	87.72%
LSTM	82.39%	91.70%
CNN+LSTM	84.45%	92.79%
CNN$_{fix}$+LSTM	85.54%	93.75%
EigenGait	-	78.97%
Wavelet	78.55%	77.37%
Fourier	92.70%	61.86%

Table VII shows the authentication results obtained by four deep-learning-based methods, i.e., LSTM, CNN, 'CNN+LSTM' and 'CNN$_{fix}$+LSTM', and three traditional methods, i.e., EigenGait, Wavelet and Fourier. Note that, the Dataset #5 and Dataset #6 are constructed on the same 118 subjects and the same samples. The only difference is that, the input data have been aligned in two different manners. Exactly, the samples are aligned in horizontal for Dataset #5 and in vertical for Dataset #6. First, we have to figure out which data-alignment manner can get better authentication performance.

It can be seen from Table VII, for the deep-learning based methods, results obtained on vertically aligned data are much better than that on horizontally aligned data. The improvements are about 9%, 9%, 8% and 8% for the CNN, LSTM, 'CNN+LSTM' and 'CNN$_{fix}$+LSTM', respectively. The possible reason is that, a pair of samples aligned vertically are aligned along the time. Since the input data are time-series, the data aligned along the time will facilitate the comparison, for both the LSTM and the 1-D CNN. It can also be observed that, LSTM achieves better results than CNN. It simply indicates that the LSTM can better handling time-series data by associating the bypast and upcoming signals for feature learning and state prediction. In addition to this, the CNN and LSTM are found to be complementary to further improve the performance. In Table VII, 'CNN+LSTM' and 'CNN$_{fix}$+LSTM' obtained significant higher performance than the stand-alone CNN or LSTM. It indicates the effectiveness of the combination strategy. Moreover, 'CNN$_{fix}$+LSTM' shows superior performance over 'CNN+LSTM'. The possible reason is that, the CNN block is harder to train in the relatively more complex 'CNN+LSTM' network than in the stand-alone CNN network. The dataset

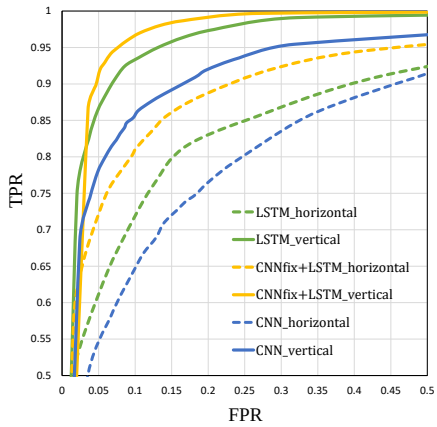


Fig. 8. ROC curves of six deep-learning authentication methods.

collected on 118 subjects can produce good results, but is still not enough for training a satisfactory and general model. Thus, 'CNN+LSTM' trained from scratch is more likely to get overfitting than 'CNN$_{fix}$+LSTM'.

For the three traditional methods, all of them obtain much lower results than the deep learning based methods, on the vertically aligned data. However, it is surprising that the Fourier-transform-based method achieves an accuracy of 92.70% on the horizontally aligned data, much higher than 61.86% on the vertically aligned data. It demonstrates the potential of frequency-domain transformation in effectively capturing the discriminative characteristics of the concatenated gait time-series. The reason is that, feature vector constructed on Fourier transform is to sample the coefficients of the frequencies from low to high in the frequency domain. For horizontally aligned data, feature vectors can be constructed on the fused signal. While for vertically aligned data, feature vectors can only be constructed independently. The independent features would be less capable of capturing the discriminative characteristics. For the Wavelet-based method, the results are very close on the horizontally- and vertically- aligned data. It indicates the channel-combination method on a pair of samples has little effect on the components of continuous wavelet transform at different scales.

Fig. 8 shows the ROC curves obtained by three deep-learning authentication methods using two different data-alignment strategies. The three deep learning methods are plotted in three different colors, while the data difference is indicated by solid line and dash line. It can be clearly observed that, the deep learning methods obtain much higher performance on vertically aligned data than on horizontally aligned ones. It indicates that the inertial data vertically aligned can better represent the relation of two samples. In the vertically aligned manner, gait phases of two input samples that are close in the time space will also be close in the spatial space, which can facilitate the discrimination of their difference.

VI. Conclusion

In this paper, gait recognition using smartphones in the wild was studied. A hybrid deep learning method was proposed to seamlessly combine the DCNN and DRNN for robust inertial gait feature representation. In gait data collection, the smartphones were used in a condition of unconstrained, and information of when, where, and how the user walks was totally unknown. A fully convolutional neural network was presented to partition the inertial data into the walking session and the non-walking session, where hierarchical convolutional features are fused together for accurate semantic segmentation. Then, a CNN with one-dimension kernels was used to transform the input time-series into convolutional feature maps, which were then carefully rearranged as time-series feature maps and fed into an LSTM for gait feature extraction. In the experiments, the extracted features obtained by the proposed method were found to be very discriminative for person identification and authentication.

In the experiments we found that the performance on accelerometer data is generally better than that on gyroscope data, and the accelerometer data and gyroscope data can be complementary to further improve the performance. We also found that, the inertial data aligned in vertical are much more helpful than that aligned in horizontal for person authentication in the proposed deep-learning framework.To promote the research in inertia-based gait recognition, we have released the collected datasets, the codes and the trained model at https://github.com/qinnzou/.

References

[1] R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, M. Ristori, S. Ricerche, and F. Roli, "Fusion of multiple clues for photo-attack detection in face recognition systems," in *IJCB*, 2011.
[2] S. Kim, S. Yu, K. Kim, Y. Ban, and S. Lee, "Face liveness detection using variable focusing," in *ICB*, 2013, pp. 1–6.
[3] M. Nixon, T. Tan, and R. Chellapppa, "Human identification based on gait," *Springer Science + Business Media Inc.*, 2006, ch. 1.
[4] J. Zhang, J. Pu, C. Chen, and R. Fleischer, "Low-resolution gait recognition," *IEEE Trans. on SMC, Part B: Cybernetics*, vol. 40, no. 4, pp. 986–996, 2010.
[5] M. Ding and G. Fan, "Multilayer joint gait-pose manifolds for human gait motion modeling," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2413–2424, 2015.
[6] I. Rida, S. Al-maadeed *et al.*, "Robust gait recognition: a comprehensive survey," *IET Biometrics*, 2018.
[7] Q. Zou, L. Ni, Q. Wang, Q. Li, and S. Wang, "Robust gait recognition by integrating inertial and rgbd sensors," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1136–1150, 2018.
[8] D. Gafurov and E. Snekkenes, "Gait recognition using wearable motion recording sensors," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 7, 2009.
[9] F. Xu, C. Bhagavatula, A. Jaech, U. Prasad, and M. Savvides, "Gait-ID on the move: Pace independent human identification using cell phone accelerometer dynamics," in *BTAS*, 2012, pp. 8–15.
[10] N. Trung, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "Performance evaluation of gait recognition using the largest inertial sensor-based gait database," in *IAPR ICB*, 2012, pp. 360–366.
[11] Y. Zhang, G. Pan, K. Jia, M. Lu, Y. Wang, and Z. Wu, "Accelerometer-based gait recognition by sparse representation of signature points with clusters," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1864–1875, 2015.
[12] S. Sprager and M. B. Juric, "An efficient hos-based gait authentication of accelerometer data," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1486–1498, 2015.
[13] J. Kwapisz, G. Weiss, and S. Moore, "Cell phone-based biometric identification," in *BTAS*, 2010, pp. 1–7.
[14] B. Sun, Y. Wang, and J. Banda, "Gait characteristic analysis and identification based on the iphones accelerometer and gyrometer," *Sensors*, vol. 14, no. 9, pp. 17 037–17 054, 2014.
[15] P. Fernandez-Lopez, J. Sanchez-Casanova, P. Tirado-Martín, and J. Liu-Jimenez, "Optimizing resources on smartphone gait recognition," in *ICB*, 2017, pp. 31–36.
[16] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

[17] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015, pp. 2625–2634.

[18] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deepcrack: Learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, pp. 1–15, 2018.

[19] M. D. Addlesee, A. Jones, F. Livesey, and F. Samaria, "The orl active floor [sensor system]," *IEEE Personal Communications*, vol. 4, no. 5, pp. 35–41, 1997.

[20] B. Huang, M. Chen, P. Huang, and Y. Xu, "Gait modeling for human identification," in *ICRA*, 2007, pp. 4833–4838.

[21] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers." in *AAAIW*, 2016.

[22] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1553–1567, 2006.

[23] B. Shrestha, D. Ma, Y. Zhu, H. Li, and N. Saxena, "Tap-wave-rub: Lightweight human interaction approach to curb emerging smartphone malware," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2270–2283, 2015.

[24] R. E. Mayagoitia, A. V. Nene, and P. H. Veltink, "Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems," *Journal of biomechanics*, vol. 35, no. 4, pp. 537–542, 2002.

[25] H. Zhao, Z. Wang, S. Qiu, Y. Shen, L. Zhang, K. Tang, and G. Fortino, "Heading drift reduction for foot-mounted inertial navigation system via multi-sensor fusion and dual-gait analysis," *IEEE Sensors Journal*, 2018.

[26] J. Frank, S. Mannor, and D. Precup, "Activity and gait recognition with time-delay embeddings." in *AAAI*, 2010.

[27] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *ICASSP*, 2005, pp. 973–976.

[28] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers," in *BTHI*, 2005.

[29] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor," *Journal of Computers*, vol. 1, no. 7, pp. 51–59, 2006.

[30] D. Gafurov, E. Snekkenes, and P. Bours, "Spoof attacks on gait authentication system," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 491–502, 2007.

[31] ——, "Gait authentication and identification using wearable accelerometer sensor," in *AIATW*, 2007, pp. 220–225.

[32] R. Liu, Z. Duan, J. Zhou, and M. Liu, "Identification of individual walking patterns using gait acceleration," in *ICBBE*, 2007.

[33] R. Liu, J. Zhou, M. Liu, and X. Hou, "A wearable acceleration sensor system for gait recognition," in *ICIEA*, 2007, pp. 2654–2659.

[34] G. Trivino, A. Alvarez-Alvarez, and G. Bailador, "Application of the computational theory of perceptions to human gait pattern recognition," *Pattern Recognition*, vol. 43, no. 7, pp. 2572–2581, 2010.

[35] M. Derawi, P. Bours, and K. Holien, "Improved cycle detection for accelerometer based gait authentication," in *IIHMSP*, 2010.

[36] S. Sprager and M. B. Juric, "Inertial sensor-based gait recognition: a review," *Sensors*, vol. 15, no. 9, pp. 22 089–22 127, 2015.

[37] N. Trung, Y. Makihara, H. Nagahara, R. Sagawa, Y. Mukaigawa, and Y. Yagi, "Phase registration in a gallery improving gait authentication," in *IJCB*, 2011.

[38] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "Orientation-compensative signal registration for owner authentication using an accelerometer," *IEICE Transactions on Information and Systems*, vol. 97, no. 3, pp. 541–553, 2014.

[39] T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication," *Pattern Recognition*, vol. 47, no. 1, pp. 228–237, 2014.

[40] Y. Zhong and Y. Deng, "Sensor orientation invariant mobile gait biometrics," in *IJCB*, 2014, pp. 1–8.

[41] A. H. Johnston and G. M. Weiss, "Smartwatch-based biometric gait recognition," in *BTAS*, 2015, pp. 1–6.

[42] J. Le Moing and I. Stengel, "The smartphone as a gait recognition device impact of selected parameters on gait recognition," in *ICISSP*, 2015.

[43] H. Abujrida, E. Agu, and K. Pahlavan, "Smartphone-based gait assessment to infer parkinson's disease severity using crowdsourced data," in *HI-POCT*, 2017, pp. 208–211.

[44] J. Juen, Q. Cheng, V. Prieto-Centurion, J. A. Krishnan, and B. Schatz, "Health monitors for chronic disease by gait analysis with mobile phones," *Telemedicine and e-Health*, vol. 20, no. 11, pp. 1035–1041, 2014.

[45] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1961–1974, 2015.

[46] A. Ferreira, G. Santos, A. Rocha, and S. Goldenstein, "User-centric coordinates for applications leveraging 3-axis accelerometer data," *IEEE Sensors Journal*, vol. 17, no. 16, pp. 5231–5243, 2017.

[47] J. Lu, G. Wang, and P. Moulin, "Human identity and gender recognition from gait sequences with arbitrary walking directions," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 51–61, 2014.

[48] P. Chattopadhyay, S. Sural, and J. Mukherjee, "Frontal gait recognition from incomplete sequences using rgb-d camera," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1843–1856, 2014.

[49] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep cnns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 209–226, 2017.

[50] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[51] J. Man and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[52] L. Lee and W. E. L. Grimson, "Gait analysis for recognition and classification," in *AFGR*, 2002, pp. 155–162.

[53] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio, "Full-body person recognition system," *Pattern Recognition*, vol. 36, no. 9, pp. 1997–2006, 2003.

[54] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *IJCAI*, 2011.

[55] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213–2220, 2008.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[57] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *MCAS*, 2014, pp. 197–205.

[58] E. P. Ijjina and C. K. Mohan, "One-shot periodic activity recognition using convolutional neural networks," in *ICMLA*, 2014, pp. 388–391.

[59] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, pp. 103–110, 2017.

[60] W. Yuan and L. Zhang, "Gait classification and identity authentication using cnn," in *Asian Simulation Conference*, 2018, pp. 119–128.

[61] Y. Zhao and S. Zhou, "Wearable device-based gait recognition using angle embedded gait dynamic images and a convolutional neural network," *Sensors*, vol. 17, no. 3, p. 478, 2017.

[62] N. Takemura, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "On input/output architectures for convolutional neural network-based cross-view gait recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[63] M. Gadaleta and M. Rossi, "Idnet: Smartphone-based gait recognition with convolutional neural networks," *Pattern Recognition*, vol. 74, pp. 25–37, 2018.

[64] C. Li, X. Min, S. Sun, W. Lin, and Z. Tang, "Deepgait: a learning deep convolutional representation for view-invariant gait recognition using joint bayesian," *Applied Sciences*, vol. 7, no. 3, p. 210, 2017.

[65] F. M. Castro, M. J. Marín-Jiménez, N. Guil, and N. P. de la Blanca, "Automatic learning of gait signatures for people identification," in *ICANN*, 2017, pp. 257–270.

[66] T. Wolf, M. Babaee, and G. Rigoll, "Multi-view gait recognition using 3d convolutional neural networks," in *ICIP*, 2016.

[67] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[68] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[69] L. Wu, C. Shen, and A. Hengel, "Convolutional lstm networks for video-based person re-identification," *arXiv:1606.01609*, 2016.

[70] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *IJCAI*, 2016.

[71] S. Yu, H. Chen, E. B. G. Reyes, and P. Norman, "Gaitgan: invariant gait feature extraction using generative adversarial networks," in *CVPRW*, 2017, pp. 30–37.

[72] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *ICASSP*, 2016.
[73] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.
[74] J. Mäntyjärvi, M. Lindholm, E. Vildjiounaite, S.-M. Mäkelä, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers." in *ICASSP*, 2005, pp. 973–976.