PLOS ONE

RESEARCH ARTICLE

Explainable gait recognition with prototyping encoder–decoder

Jucheol Moon[1], **Yong-Min Shin**[2], **Jin-Duk Park**[2], **Nelson Hebert Minaya**[1], **Won-Yong Shin**[2]*, **Sang-Il Choi**[3]*

1 Department of Computer Engineering and Computer Science, California State University, Long Beach, CA, United States of America, **2** School of Mathematics and Computing (Computational Science and Engineering), Yonsei University, Seoul, Republic of Korea, **3** Department of Computer Science and Engineering, Dankook University, Yongin-si, Gyeonggi-do, Republic of Korea

* wy.shin@yonsei.ac.kr (WYS); choisi@dankook.ac.kr (SC)

OPEN ACCESS

Citation: Moon J, Shin Y-M, Park J-D, Minaya NH, Shin W-Y, Choi S-I (2022) Explainable gait recognition with prototyping encoder–decoder. PLoS ONE 17(3): e0264783. https://doi.org/10.1371/journal.pone.0264783

Editor: Thippa Reddy Gadekallu, Vellore Institute of Technology: VIT University, INDIA

Received: September 16, 2021

Accepted: February 16, 2022

Published: March 11, 2022

Data Availability Statement: This dataset was approved by the Institutional Review Board of California State University Long Beach (IRB No. 21-091); thus it is restricted to commercial use and is available only for acknowledged research for non-commercial purposes. All relevant data are available on the following website with the approval of CSULB IRB above. https://github.com/MLPA-DKU/3L_Laps_dataset.git.

Funding: This research was supported by the National Research Foundation of Korea (NRF) grants (No. 2021R1A2C3004345 and No.

Abstract

Human gait is a unique behavioral characteristic that can be used to recognize individuals. Collecting gait information widely by the means of wearable devices and recognizing people by the data has become a topic of research. While most prior studies collected gait information using inertial measurement units, we gather the data from 40 people using insoles, including pressure sensors, and precisely identify the gait phases from the long time series using the pressure data. In terms of recognizing people, there have been a few recent studies on neural network-based approaches for solving the open set gait recognition problem using wearable devices. Typically, these approaches determine decision boundaries in the latent space with a limited number of samples. Motivated by the fact that such methods are sensitive to the values of hyper-parameters, as our first contribution, we propose a new network model that is less sensitive to changes in the values using a new *prototyping encoder–decoder* network architecture. As our second contribution, to overcome the inherent limitations due to the lack of transparency and interpretability of neural networks, we propose a new module that enables us to analyze which part of the input is relevant to the overall recognition performance using *explainable* tools such as sensitivity analysis (SA) and layer-wise relevance propagation (LRP).

1 Introduction

1.1 Background

Human gait, i.e., the way in which people walk, is sufficiently unique to distinguish one individual from another. Gait information has been utilized for diverse applications such as disease diagnosis [1] and biometric authentication [2]. Compared to other biometric authentication methods, gait recognition is advantageous in that it is robust against impersonation attacks, not necessarily requiring vision sensors or physical contacts with sensing devices to collect data [3].

2021R1A2B5B01001412) funded by the Korea government (MSIT) and by the Republic of Koreas MSIT (Ministry of Science and ICT), under the High-Potential Individuals Global Training Program) (No. 2020-0-01463) supervised by the IITP (Institute of Information and Communications Technology Planning Evaluation).

Competing interests: The authors have declared that no competing interests exist.

A gait recognition framework comprises two core components: data acquisition devices and data analysis algorithms. It captures representational data of the gait and identifies individuals by classifying such data using different algorithms. To be more precise, the gait information can be captured by using vision sensors, pressure sensors, and inertial measurement units (IMU); then, the captured data are classified using the linear discriminant analysis (LDA), *k*-nearest neighbor (*k*-NN), hidden markov model (HMM), support vector machine (SVM), convolutional neural network (CNN), or the combinations thereof [3]. A recognition problem can be categorized into two types: the closed set problem and the open set problem. Whereas the closed set recognition tests samples of classes known from training, the open set recognition deals with incomplete knowledge given at the time of training and tests not only known but also unknown classes [4], which is a more challenging task. While the majority of gait recognition frameworks have focused on the closed set recognition, few approaches have addressed the open set recognition in the literature [5].

1.2 Main contributions

Fig 1 shows the motivation of the objectives of our study. We assume a wireless environment in which all participants wear shoes with sensor-equipped insoles that can communicate wirelessly. This is due to not only an ease of data acquisition but also an availability of high-quality sensors. Under such a circumstance, as an aspect of data acquisition, the time series of each individual's gait is captured by pressure sensors, a 3D-axis accelerometer, and a 3D-axis gyroscope installed in the insoles of the participants' shoes. Because the data are collected by the insoles, different than other publicly available datasets [6, 7], pressure values between the foot and the ground can be measured in addition to the IMUs. Using the pressure values, the continuous gait data are segmented into separate unit steps for the gait recognition framework to perform in a more efficient and effective manner along with the human walking cycles [8], which consist of a stance phase and a swing phase [9]; the stance phase is the time when a foot is on the ground, and the swing phase is the entire time when a foot is in the air. Using the fact

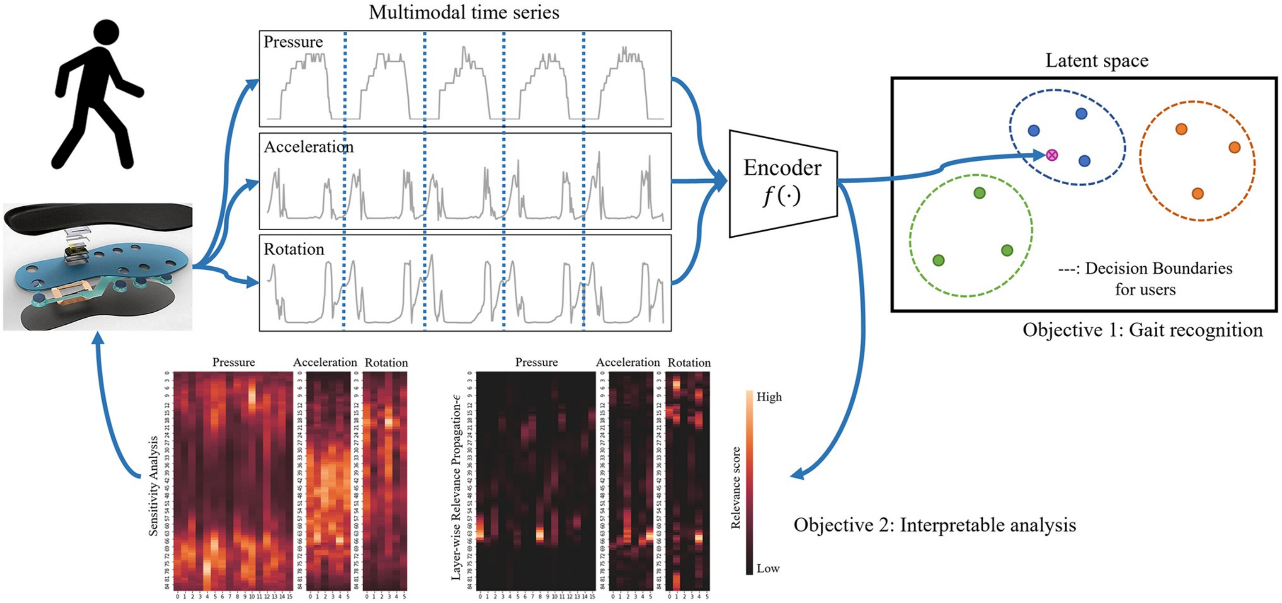

Fig 1. Illustration of the motivation and the objectives. Our study is to recognize a set of users from their gait patterns using a encoder network and to provide an interpretable analysis of the network using the XAI method.

https://doi.org/10.1371/journal.pone.0264783.g001

that the pressure values should be zero during the swing phase, the continuous data are split into unit steps. When forming unit steps, Gaussian smoothing is applied to reduce potential errors [8] such that pressure sensors sporadically show non-zeros during the swing phase [10]. To utilize the merit of using pressure values, we collected the data from 40 participants using the insoles instead of using public database contains only acceleration and gyroscope data.

As our first main contribution, we propose an encoder–decoder network model with multiple 1D convolutional layers. The encoder maps the multimodal unit steps into embedding vectors in a latent space, and the decoder reconstructs the unit steps from the embedding vectors. To train this network, a linear combination of two loss functions is used, i.e., $L = L_{triplet} + \lambda L_{proto}$ where $\lambda \geq 0$. Here, the first loss function, denoted by $L_{triplet}$, is based on the triplet loss [11], and the second loss function, denoted by L_{proto}, similarly follows that of the denoising autoencoder [12]. The $L_{triplet}$ widens the distance between the embedding vectors of the heterogeneous unit steps while narrowing the distance between the embedding vectors of the homogeneous unit steps in the latent space. For homogeneous unit steps, we compute their prototype by averaging out over the unit step data; then, the L_{proto} forces the encoder–decoder network to minimize the difference between the homogeneous unit steps and their prototype. To develop and evaluate our gait recognition system to effectively address an under-explored *open set* recognition problem, we split the data into *training*, *known test*, and *unknown test* sets.

Once the encoder–decoder network is trained for the embedding vectors, the one-class support vector machine (OSVM) [13] is used to classify the unit steps. A *few* unit steps of each person in the *known test* set are randomly selected. Using the embedding vectors of the selected unit steps, OSVM is trained to compute a decision boundary for each subjects in the *known test* set. The classifiers are thereby capable of identifying whether a unit step belongs to any of the known classes. During the test phase of the system, when an unseen unit step of a subject is given, it is first mapped into an embedding vector using the encoder. The embedded vector is then examined if it is within the decision boundary of the class whose centroid is the closest among the known classes. Otherwise, it is rejected.

On the other hand, it is worth noting that the use of such neural network-based models has generally been regarded as a black-box because of the lack of transparency and interpretability [14]. The highly non-linear nature of neural networks hinders our attempt to understand the decision-making process of such models. To overcome this barrier, studies on *explainable* artificial intelligence (XAI) have emerged [15, 16], allowing transparency and interpretability to be improved in neural networks. Improved transparency offers context to the model decision, which thus leads to several benefits. First, XAI can build trust for users of a given model. Second, the interpretation itself can be used as extra information to obtain a more complete understanding [14].

The application of XAI for gait data analysis is still largely under-explored despite its potential. Several recent studies have applied XAI to the closed set gait recognition problem [17, 18]. Our study aims to utilize XAI to understand the process of the open set gait recognition. In this study, as our second main contribution, we incorporate two well-known XAI tools, namely sensitivity analysis (SA) [19] and layer-wise relevance propagation (LRP) [20], into our gait recognition framework. Each method calculates *attribution maps*, where the values indicate the importance of the input when the underlying model returns the output.

To interpret the encoder, we apply SA and LRP to the embedding vectors. However, unlike the closed set recognition models, we take the expectation of all attribution maps calculated from each dimension of the embedding space because the embedding space has no explicit interpretation. Finally, by averaging out the attribution maps over all training subjects, it is possible to obtain a common attribution map that represents important parts of the entire *training* set for gait recognition.

The main contributions of this study are summarized as follows:

- Using a combination of the triplet and prototype loss functions allows our encoder–decoder network to be more robust and less dependent on the values of hyper-parameters;
- XAI approaches are shown to obtain insights from a human interpretable analysis of a neural network-based open set gait recognition model, which demonstrates how high and low relevant parts of the data affect the accuracy of the recognition.

The remainder of this paper is organized as follows. In Section 2, we summarize significant studies that are related to our work. In Section 3, we describe the dataset for gait recognition. Section 4 explains our proposed methods. Experimental results are provided in Section 5. Finally, we summarize the paper with some concluding remarks in Section 6.

2 Related work

The method that we propose in this paper is related to two broader areas of research, namely gait recognition and explainable neural networks.

2.1 Gait recognition

The use of vision sensors led to the beginning of gait recognition analyses [21]; follow-up studies have actively been carried out in the literature [22, 23]. Despite challenging conditions for collecting data in vision-based recognition (e.g., requiring only the subject of interest in the video sequences), the accuracy of gait recognition based on these vision-based approaches is insufficiently high and yet unstable depending on the viewpoint and orientation of the sensing devices [24]. To overcome these obstacles, not only subjects in video sequences were segmented and individually tracked [24], but also 3D construction and view transformation models were used [25]. A view-adaptive mapping approach for gait recognition was also developed in [26] to alleviate the free-view gait recognition problem in which the view angle is often unknown, dynamically changing, or does not belong to any predefined views.

In addition to such vision-based approaches, pressure sensors and IMUs have been broadly used to collect data in recent gait recognition analyses. IMUs typically consist of an accelerometer, a gyroscope, and a magnetometer. For example, gait information was gathered from IMUs attached to multiple parts of each participant's body [27], and then the individuals were identified using a CNN-based predictive model [28]. Later, pressure sensors and IMUs installed in wearable devices, e.g., smartphones, fitness trackers, or shoe insoles [29], were used. In a study on smartphone-based gait recognition [30], data from IMUs in smartphones were analyzed using a mixed model of CNN and SVM [31]. In another study, null space LDA was applied to analyze gait data from pressure sensors and accelerometers placed in shoe insoles [32]. These methods, however, were limited in the sense of placing multiple sensors on various parts of the body, taking a long period of time to gather data, or showing insufficient performance of identification. The list of related studies is summarized in Table 1. Except for [8, 32], all the related studies collected data using only IMU sensors. On the other hand, we collected the data using insoles, where both IMU sensors and pressure sensors are installed within them. The distinguishing feature of our research from related studies is the use of pressure sensors with IMU sensors. Note that the pressure sensor data are useful due to the fact that not only they have not been taken into account in the other studies, but also the time series is split into small fragments corresponding to the phase of human gait. The detailed description is presented in Section 3.

As for an open set gait recognition problem, a few studies have been conducted, each of which was performed differently. For example, a CNN-based classification model was

Table 1. List of the related work on gait recognition.

Authors	Year	Sensor position	Sensor types
Luo et al. [33]	2020	Trunk, wrist, thighs, shanks	Acceleration, gyroscope, magnetic, orientation
Moon et al. [8]	2020	Foot	Pressure, acceleration, gyroscope
Choi et al. [32]	2019	Foot	Pressure, acceleration
Weiss et al. [34]	2019	Pants pocket, hand	Acceleration, gyroscope
Gadaleta et al. [30]	2018	Pants pocket	Acceleration, gyroscope
Al Kork et al. [35]	2017	Upper pocket, wrist, pants pocket, bag, leg, hand	Acceleration, gyroscope
Chereshnev et al. [36]	2017	Foot, shanks, thighs	Acceleration, gyroscope
Subramanian et al. [37]	2015	Pocket, holster	Acceleration, gyroscope, magnetic, orientation
Ngo et al. [38]	2014	Inside backpack	Acceleration, orientation
Anhuita et al. [39]	2013	Waist	Acceleration, gyroscope
Frank et al. [40]	2013	Pocket	Acceleration
Reiss et al. [41]	2012	Chest, wrist, ankle	Acceleration, gyroscope, magnetic
Zhang et al. [42]	2012	Hip	Acceleration
Altun et al. [43]	2010	Knees, chest, wrists	Acceleration, gyroscope, magnetic
Bächlin et al. [44]	2010	Shank, thigh, lower back	Acceleration
Gafurov et al. [45]	2010	Ankle	Acceleration

https://doi.org/10.1371/journal.pone.0264783.t001

designed to have a softmax output layer including a 'not recognized' class for unknown subjects at the time of training [46]. However, this approach is not scalable since the network model needs to be trained whenever a new subject is added to the system. In another study [30], a framework based on CNN and OSVM was used, but the system requires about a hundred unit steps to train the OSVM while not being evaluated with samples of truly unknown subjects. More recently, the open set problem was successfully handled in [47], proposing a framework with an ensemble model of CNN and recurrent neural network (RNN) along with the OSVM algorithm. It requires only a few unit steps to train OSVM while being evaluated with unseen samples of both known and unknown subjects to the OSVM-based classifier.

2.2 Explainable neural networks

Although a wide range of studies on XAI have been carried out in various domains [48–50], *post-hoc* methods, operating on the underlying model to be interpreted after the training has ended, have received considerable attention due to ease of implementation, as in our study. As one of popular post-hoc methods, SA measures the gradient values from the output to the input [19]. As another post-hoc method, LRP redistributes the scores of the output layer back to the input, rather than the gradient values [20]; this relies on the activation values during the feedforward process to determine how the values of each layer should be distributed. Other XAI methods have also been developed. Based on two axioms for attribution maps, integrated gradients [51] calculates the average gradient while following a linear path from a baseline (usually having the zero input). DeepLIFT [52] extended the LRP method by taking into account the activation of the baseline as a reference. Besides, in [53], the LRP method was applied to predict the category of text documents using standard machine learning models such as CNN.

To evaluate attribution maps, region perturbation was introduced in [54], where occluding parts of the input are shown with respect to relevance scores. According to the method in [54], the occluded parts were replaced by randomly sampled values. As an alternative, those parts were substituted with zero values [55]. In our study, we make some modifications in such a way that the absolute values of the relevance scores are used.

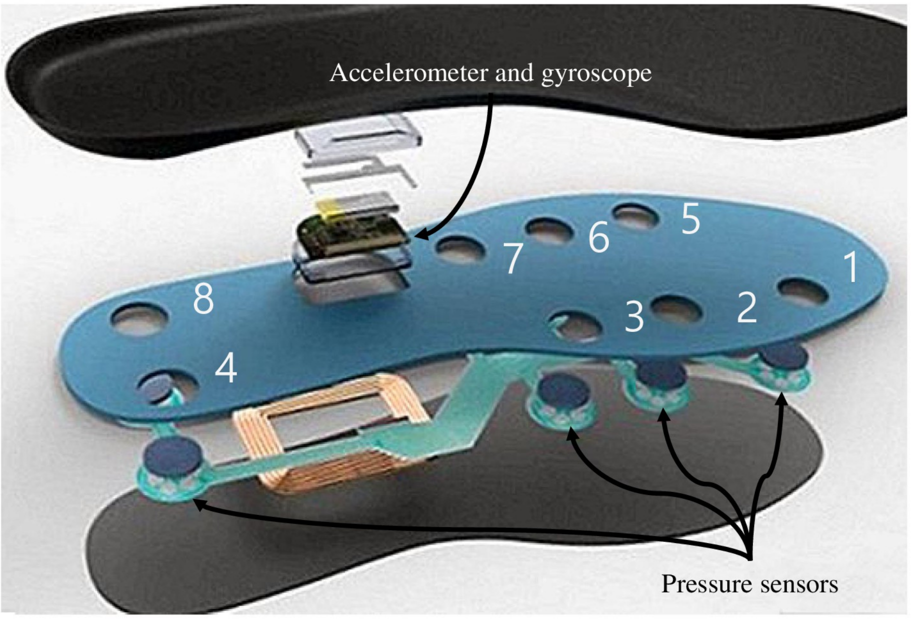

Fig 2. The insole used to collect the gait data.

https://doi.org/10.1371/journal.pone.0264783.g002

3 Data description and prototyping

To collect the gait information of the subjects, we utilized a commercial shoe insole, FootLogger [56], as illustrated in Fig 2. Eight pressure sensors, one 3D-axis accelerometer, and one 3D-axis gyroscope were installed in the insole. The pressure sensor gauges the pressure at one of three levels, and the accelerometer as well as the gyroscope measure the acceleration and rotation, respectively, leading to 2^{16} levels. While the subjects walked, the insole recorded data at every 0.01 second.

The pressure sensors report non-zero values when a foot is on the ground and show zero values otherwise. Using this property, as in [8], the original time series is converted into a series of unit steps in a fixed length, each of which includes data for one walking cycle. We partially adapted the notation in [47]. The i^{th} data of subject $id = a$ for sensing modality m are denoted by $\mathbf{s}^{m}_{i,a}$, where $m \in \mathcal{M} = \{p_{l1}, \cdots, p_{l8}, p_{r1}, \cdots, p_{r8}, a_{lx}, \cdots, a_{rz}, r_{lx}, \cdots, r_{rz}\}$. $\mathcal{M}$ is a set of all modalities. For example, p_{l1} denotes the pressure from the pressure sensor $id = 1$ in the left foot insole; a_{lx} denotes the acceleration along the x-axis in the left foot insole; and r_{rz} denotes the rotation along the z-axis in the right foot insole. We define $\mathbf{s}^{pre}_{i,a}$, $\mathbf{s}^{acc}_{i,a}$, $\mathbf{s}^{rot}_{i,a}$, and $\mathbf{s}_{i,a}$ as follows:

$$\begin{aligned}
\mathbf{s}^{pre}_{i,a} &= [\mathbf{s}^{p_{l1}}_{i,a}, \cdots, \mathbf{s}^{p_{l8}}_{i,a}, \mathbf{s}^{p_{r1}}_{i,a}, \cdots, \mathbf{s}^{p_{r8}}_{i,a}] \\
\mathbf{s}^{acc}_{i,a} &= [\mathbf{s}^{a_{lx}}_{i,a}, \mathbf{s}^{a_{ly}}_{i,a}, \mathbf{s}^{a_{lz}}_{i,a}, \mathbf{s}^{a_{rx}}_{i,a}, \mathbf{s}^{a_{ry}}_{i,a}, \mathbf{s}^{a_{rz}}_{i,a}] \\
\mathbf{s}^{rot}_{i,a} &= [\mathbf{s}^{r_{lx}}_{i,a}, \mathbf{s}^{r_{ly}}_{i,a}, \mathbf{s}^{r_{lz}}_{i,a}, \mathbf{s}^{r_{rx}}_{i,a}, \mathbf{s}^{r_{ry}}_{i,a}, \mathbf{s}^{r_{rz}}_{i,a}] \\
\mathbf{s}_{i,a} &= [\mathbf{s}^{pre}_{i,a}, \mathbf{s}^{acc}_{i,a}, \mathbf{s}^{rot}_{i,a}].
\end{aligned}$$

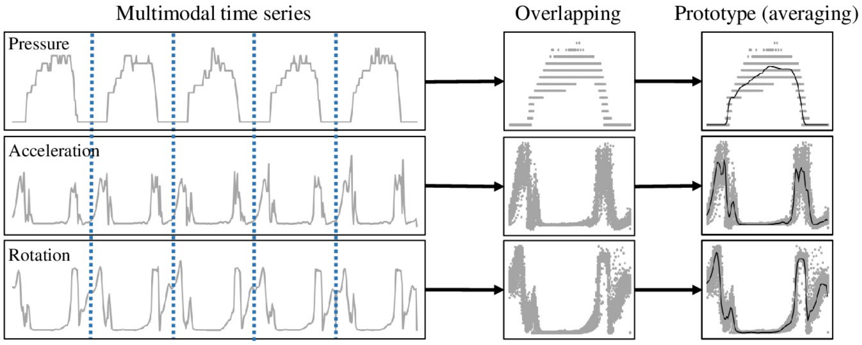

Fig 3. Illustration of computing the prototype of each sensing modality for a subject. The prototypes (bold solid curves in the rightmost figures) of a subject are computed by averaging over all unit steps. For brevity, the L_2 norms of $s^{pre}_{i,a}$, $s^{acc}_{i,a}$, and $s^{rot}_{i,a}$ are depicted.

https://doi.org/10.1371/journal.pone.0264783.g003

We call $\mathbf{s}_{i,a}$ by the i^{th} *unit step* of subject $id = a$. In addition, for sensing modality m and subject $id = a$, the *prototype* is defined as follows:

$$\mathbf{c}^m_a = \frac{1}{q}\sum_{i=1}^{q}\mathbf{s}^m_{i,a}, \tag{1}$$

where q is the number of unit steps of subject $id = a$. Then, we define the prototypes for three types of sensors by

$$\begin{aligned}
\mathbf{c}^{pre}_a &= [\mathbf{c}^{p_{l1}}_a, \cdots, \mathbf{c}^{p_{l8}}_a, \mathbf{c}^{p_{r1}}_a, \cdots, \mathbf{c}^{p_{r8}}_a] \\
\mathbf{c}^{acc}_a &= [\mathbf{c}^{a_{lx}}_a, \mathbf{c}^{a_{ly}}_a, \mathbf{c}^{a_{lz}}_a, \mathbf{c}^{a_{rx}}_a, \mathbf{c}^{a_{ry}}_a, \mathbf{c}^{a_{rz}}_a] \\
\mathbf{c}^{rot}_a &= [\mathbf{c}^{r_{lx}}_a, \mathbf{c}^{r_{ly}}_a, \mathbf{c}^{r_{lz}}_a, \mathbf{c}^{r_{rx}}_a, \mathbf{c}^{r_{ry}}_a, \mathbf{c}^{r_{rz}}_a].
\end{aligned}$$

A conceptual diagram of computing the prototype is depicted in Fig 3.

4 Proposed methods

We assume a wireless environment in which all participants wear shoes with sensor-equipped insoles that can communicate wirelessly. The research problems are stated as follows:

- Given a set of data collected using the insoles, we aim at recognizing users using our encoder–decoder network to be more robust and less dependent on the values of hyperparameters;
- Given the trained network for the gait recognition, we aim at demonstrating how high and low relevant parts of the data affect the accuracy of the recognition.

We first present our encoder–decoder architecture for gait recognition in Subsection 4.1. We then elaborate on two types of XAI methods built upon the designed architecture in Subsection 4.2.

The study was conducted according to the guidelines of the Declaration of Helsinki and approved by the Institutional Review Board of California State University Long Beach (IRB No. 21-091).

4.1 Gait recognition

In this subsection, we describe our proposed encoder–decoder network architecture and a few-shot learning approach for gait recognition. In the previous work [47], an ensemble model of CNN and RNN with the triplet loss function showed the recognition accuracy about 93%, which is however quite sensitive to the values of hyper-parameters. This motivates us to propose the encode–decoder network with the combination of the prototype and triplet loss function to overcome the issue.

4.1.1 Network architecture. We propose an encoder–decoder network architecture alongside two loss functions. The encoder $f(\cdot)$ maps a unit step to an embedding vector in a latent space, and the decoder $g(\cdot)$ maps an embedding vector to a prototype. The encoder $f(\cdot)$ includes three identical sub-encoders $f_{sub}(\cdot)$, which consist of three one-dimensional (1D) convolutional layers with 32, 64, and 128 filters and a flattened layer in order. The last flattened layers of three sub-encoders are fully connected to a dense layer with 256 units, and the dense layer is fully connected to another dense layer with 128 units, which is the output of the encoder. Similarly as in the encoder, the decoder $g(\cdot)$ includes one dense layer with 256 units and three identical sub-decoders $g_{sub}(\cdot)$, which consist of the same layers as those in the sub-encoders in reverse order. Although the layouts of the sub-encoders or sub-decoders are identical, their parameters are independently trained using different sensing modalities, including pressure, acceleration, and rotation. The network architecture is depicted in Fig 4.

In the encoder–decoder network architecture, we take the middle dense layer with 128 units as the output of the encoder. The encoder maps unit steps of $\mathbf{s} \triangleq [\mathbf{s}_{i,a}^{pre}, \mathbf{s}_{i,a}^{acc}, \mathbf{s}_{i,a}^{rot}]$ to embedding vectors $\mathbf{v}$:

$$f(\mathbf{s}_{i,a}^{pre}, \mathbf{s}_{i,a}^{acc}, \mathbf{s}_{i,a}^{rot}) = f(\mathbf{s}) = \mathbf{v}, \tag{2}$$

where the dimension of embedding vectors is 128 (i.e., $f(\cdot) \in \mathbb{R}^{128}$) and the vectors are normalized to 1 (i.e., $||f(\cdot)||_2 = 1$). Hereafter, to simplify notations, $\mathbf{s}_{i,a}^{pre}$, $\mathbf{s}_{i,a}^{acc}$, and $\mathbf{s}_{i,a}^{rot}$ will be written as $\mathbf{s}^{pre}$, $\mathbf{s}^{acc}$, and $\mathbf{s}^{rot}$, respectively, if dropping the subscript (i, a) does not cause any confusion.

Let $\mathbf{s}_{i,a}$ and $\mathbf{s}_{j,a}$ $(i \neq j)$ be two unit steps of subject $id = a$, and let $\mathbf{s}_{k,b}$ be a unit step of subject $id = b$. Similarly as in the triplet loss [11], our multimodal triplet loss is defined as follows:

$$L_{triplet} = ||\mathbf{v}_{i,a} - \mathbf{v}_{j,a}||_2^2 - ||\mathbf{v}_{i,a} - \mathbf{v}_{k,b}||_2^2 + \alpha, \tag{5}$$

where $\mathbf{v}_{i,a} = f(\mathbf{s}_{i,a})$, $\mathbf{v}_{j,a} = f(\mathbf{s}_{j,a})$, $\mathbf{v}_{k,b} = f(\mathbf{s}_{k,b})$, and $\alpha \geq 0$ is a margin. We set $\alpha = 1.25$ in the experiments unless otherwise stated.

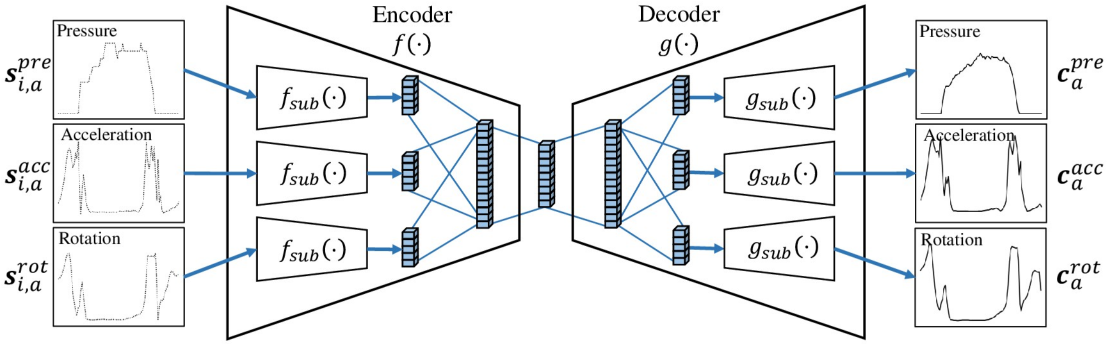

Fig 4. Illustration of the encoder–decoder architecture. The encoder and decoder include three sub-encoders and sub-decoders for multimodal sensing.

https://doi.org/10.1371/journal.pone.0264783.g004

The multimodal triplet loss attempts to ensure that the distance between two embedding vectors $\mathbf{v}_{i,a}$ and $\mathbf{v}_{j,a}$ is smaller than the distance between another pair of embedding vectors $\mathbf{v}_{i,a}$ and $\mathbf{v}_{k,b}$ for all possible triplets in the *training* set. However, an encoder with the triplet loss function can be severely distorted when the variation of the unit steps of one subject is large. To alleviate this drawback of the traditional triplet loss function, we propose an encoder–decoder architecture with a *prototyping loss*. This is similar to the denoising autoencoders [12], the center loss encoder [57], or the variational prototyping encoder [58]; however, the proposed method does not corrupt the original data, does not compute the loss in a latent space, and does not require additional prototypes as input. For each sensing modality $m \in \mathcal{M}$, the prototyping loss is defined as follows:

$$L_{proto} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} ||g(f(\mathbf{s}^m_{i,a})) - \mathbf{c}^m_a||^2_2. \tag{4}$$

Before computing the loss above, both $g(f(\mathbf{s}^m_{i,a}))$ and $\mathbf{c}^m_i$ are normalized to 1, that is $||g(f(\mathbf{s}^m_{i,a}))||_2 = ||\mathbf{c}^m_i||_2 = 1$. Since the overall loss function is given by a linear combination of the multimodal triplet loss function and the prototyping loss function, it is formulated as

$$L = L_{triplet} + \lambda L_{proto}, \tag{5}$$

where $\lambda \geq 0$ is one of the hyper-parameters of the system. A conceptual diagram of the prototyping encoder–decoder architecture is illustrated in Fig 5.

4.1.2 Few-shot learning. We split the subjects into the following three groups: *training*, *known*, and *unknown* groups. For the *training* group, all unit steps of the subjects are allocated to the *training* set. For the *known* group, n unit steps are utilized to compute the centroids of the embedding vectors and to learn the decision boundaries of the subjects using the OSVM algorithm [13], where n is one of hyper-parameters of the system, and we set $n = 10$ in the experiments unless otherwise stated. Subsequently, all unit steps with the exception of n unit steps in the *known* group are allocated to the *known test* set. For the *unknown* group, all unit steps are allocated to the *unknown test* set. From now on, we design our method based on few-shot learning (see [47] and references therein).

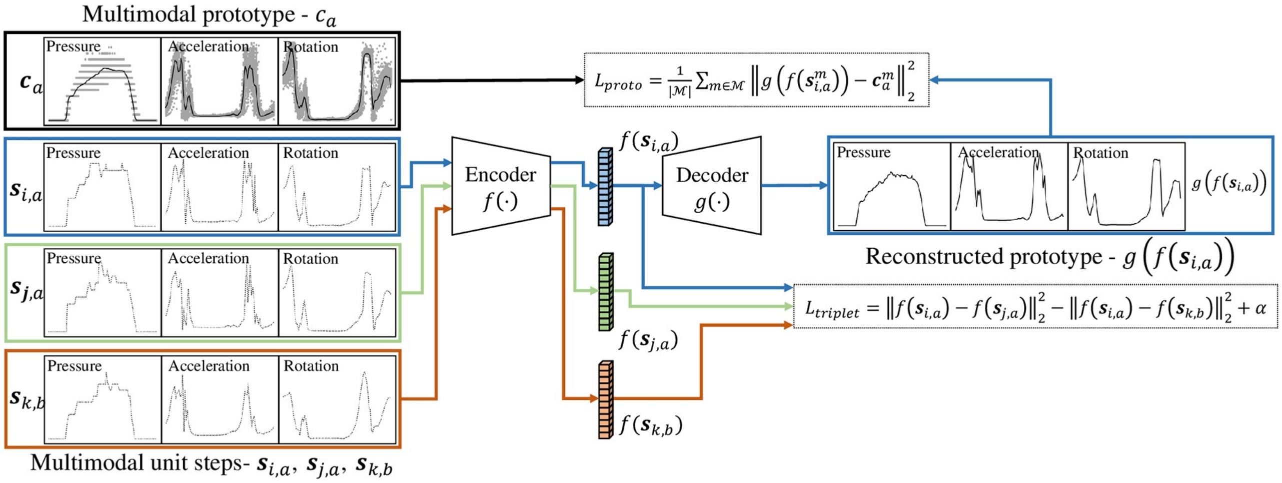
Multimodal prototype - c_a
c_a
Pressure
Acceleration
Rotation
$L_{proto} = \frac{1}{|\mathcal{M}|}\Sigma_{m\in\mathcal{M}} \left\| g\left(f\left(s^m_{i,a}\right)\right) - c^m_a \right\|^2_2$
$s_{i,a}$
Pressure
Acceleration
Rotation
Encoder
$f(\cdot)$
$f(s_{i,a})$
Decoder
$g(\cdot)$
Pressure
Acceleration
Rotation
$g\left(f(s_{i,a})\right)$
Reconstructed prototype - $g\left(f\left(s_{i,a}\right)\right)$
$s_{j,a}$
Pressure
Acceleration
Rotation
$L_{triplet} = \|f(s_{i,a}) - f(s_{j,a})\|^2_2 - \|f(s_{i,a}) - f(s_{k,b})\|^2_2 + \alpha$
$f(s_{j,a})$
$s_{k,b}$
Pressure
Acceleration
Rotation
$f(s_{k,b})$
Multimodal unit steps- $s_{i,a}$, $s_{j,a}$, $s_{k,b}$

Fig 5. Illustration of the prototyping encoder–decoder with triplet loss. The overall loss function is a linear combination of the multimodal triplet loss function and the prototyping loss function.

https://doi.org/10.1371/journal.pone.0264783.g005

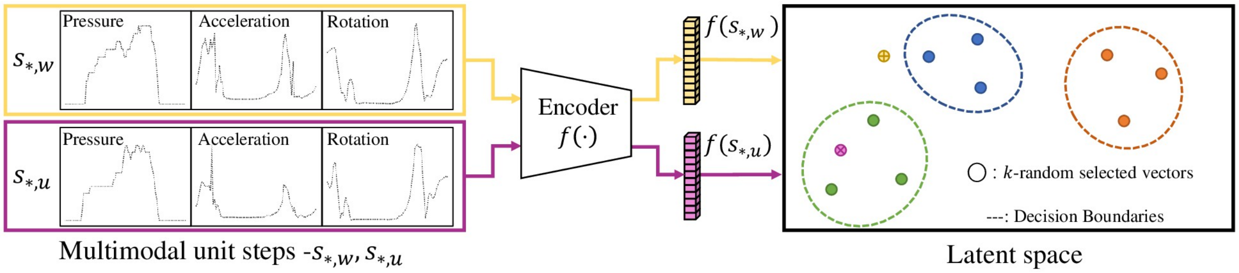

Fig 6. Illustration of gait recognition using the trained encoder. Here, unit step $\mathbf{s}_{*,u}$ is recognized as that of the "green" subject, whereas unit step $\mathbf{s}_{*,w}$ is rejected.

https://doi.org/10.1371/journal.pone.0264783.g006

Let $\mathbf{s}_{*,u}$ be a unit step of subject $id = u$ in either the *known test* or *unknown test* set. The symbol * indicates that the unit step can be any unit step of subject u. To recognize $\mathbf{s}_{*,u}$, the system first computes $\mathbf{v}_{*,u} = f(\mathbf{s}_{*,u})$ and finds the provisional subject $id = p$ such that $p = \arg\min_a ||\mathbf{D}_a - \mathbf{v}_{*,u}||_2^2$, where $\mathbf{D}_a = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_{i,a}$ for all subjects a in the *known* group. Here, the centroid $\mathbf{D}_a$ is computed using n unit steps, which are not included in the *known test* set. Next, the system discovers a decision boundary of subject p using the OSVM algorithm. Specifically, for the n embedding vectors of subject p, the algorithm uses the computed set $\{\mathbf{v}_{i,p}|1 \leq i \leq n\}$ as input and then solves the following optimization problem:

$$\begin{cases} \min_\alpha \dfrac{1}{2}\sum_i^n \sum_{i'}^n \alpha_i \alpha_{i'} \mathcal{K}(\mathbf{v}_{i,p}, \mathbf{v}_{i',p}) \\ \text{subject to}: \ 0 \leq \alpha_i \leq \dfrac{1}{\nu n}, \sum_{i=1}^n \alpha_i = 1, \end{cases} \quad (6)$$

where $\mathcal{K}(\mathbf{v}, \mathbf{v}') = e^{-\gamma||\mathbf{v}-\mathbf{v}'||_2^2}$ is a radial bias kernel function; α_i are the Lagrange multipliers; and γ and ν are the hyper-parameters of the system. The decision function of $\mathbf{v}_{*,u}$ for subject p is defined by

$$h_p(\mathbf{v}_{*,u}) = \sum_i^n \alpha_i \mathcal{K}(\mathbf{v}_{i,p}, \mathbf{v}_{*,u}) - \delta_p, \quad (7)$$

where $\delta_p = \sum_i^n \alpha_i \mathcal{K}(\mathbf{v}_{i,p}, \mathbf{v}_{h,p})$ for any h that fulfills the condition $0 < \alpha_h < \frac{1}{\nu n}$ and $1 \leq h \leq n$. Finally, the system recognizes subject u as subject p if $h_p(\mathbf{v}_{*,u}) \geq \tau$, where τ is one of the hyper-parameters of the system. A conceptual diagram of the few-shot learning-based recognition is illustrated in Fig 6, and the detailed procedure is summarized as follows:

1. Compute $\mathbf{v}_{*,u} = f(\mathbf{s}_{*,u})$;
2. Find a provisional subject $p = \arg\min_a ||\mathbf{D}_a - \mathbf{v}_{*,u}||_2^2$;
3. If $h_p(\mathbf{v}_{*,u}) \geq \tau$, then "$u$ is recognized as p".
4. Otherwise, "u is not recognized"

4.2 Explainable gait recognition

We describe two types of XAI methods for gait recognition built upon our encoder–decoder network architecture.

4.2.1 Methods for explanation. After we train the encoder $f(\cdot)$, we further implement two types of XAI methods, including SA and LRP, to gain transparency in our model and understand the decision process [14]. That is, we aim to explain our encoder $f(\cdot)$ by generating attribution maps that have the same dimensions as those in the input. Ideally, the values in an attribution map represent the importance (also known as the *relevance score*) of the input in the same position when the encoder calculates the embedding vector. For the encoder $f(\cdot)$ and given unit step input $\mathbf{s} = [\mathbf{s}^{pre}, \mathbf{s}^{acc}, \mathbf{s}^{rot}]$, our objective is to calculate an attribution map $\mathcal{A}_c(\mathbf{s})$, where $(\mathcal{A}_c(\mathbf{s}))_{ij}$ captures the degree to which the c-th component of $f(\mathbf{s})$ is relevant to the i-th row and j-th column in the input $\mathbf{s}$. Next, we would like to describe two XAI methods for gait recognition as in the following.

4.2.1.1 Sensitivity analysis (SA). One of the most widely adopted methods is to use the gradient that flows from the output to the input [19]. The gradient $\nabla_{\mathbf{s}}(f(\mathbf{s}))_c$ can be interpreted as measuring the sensitivity of the input value affecting the c-th component of the output. That is, high gradient values indicate that small deviations in the input can result in substantial changes in the model output. Each component of the gradient-based attribution map is defined as follows:

$$(\mathcal{A}_c(\mathbf{s}))_{ij} = \left| \frac{\partial (f(\mathbf{s}))_c}{\partial (\mathbf{s})_{ij}} \right|. \tag{8}$$

The gradient attribution map can be efficiently calculated using a back-propagation algorithm without any re-training of the encoder.

4.2.1.2 Layer-wise Relevance Propagation (LRP). We also take into account LRP [20] to interpret our gait recognition encoder. This method also does not require additional training of the model and efficiently calculates attribution maps. LRP calculates the relevance attribution map of given input $\mathbf{s}$ by redistributing the output value of the encoder $f(\cdot)$ back to the input. LRP starts by defining the relevance score of the final layer as the value of the output layer itself. Then, from the output layer, the method redistributes the relevance score through each layer in an iterative manner, while computing the relevance scores for each hidden layer. In general, let us denote each layer in the encoder $f(\cdot)$ as $\{l^{(0)}, \cdots, l^{(L-1)}, l^{(L)}\}$ in a sequential manner, where $l^{(0)}$ is the input layer and $l^{(L)}$ is the output layer. We also define the relevance score of neuron k in layer $l^{(H)}$ as $R_k^{(H)}$. As mentioned before, we define the relevance score in the final layer $l^{(L)}$ as the output value of the model itself, i.e., $R_k^{(L)} = (f(\mathbf{s}))_k$. To describe how the output value is redistributed back to the input layer, we consider an intermediate or the output layer $l^{(h)}$ for $h = 1, \cdots, L$. For all neurons j in layer $l^{(h)}$, the relevance score $R_j^{(h)}$ is redistributed to the neuron i in the previous layer $l^{(h-1)}$ through the following redistribution rule:

$$R_i^{(h-1)} = \sum_j \frac{x_i^{(h-1)} w_{ij}^{(h-1,h)}}{\sum_i x_i^{(h-1)} w_{ij}^{(h-1,h)}} R_j^{(h)}, \tag{9}$$

where $x_i^{(h-1)}$ is the activation value of the i-th neuron in layer $l^{(h-1)}$ and $w_{ij}^{(h-1,h)}$ is the trained weight between neuron i in layer $l^{(h-1)}$ and neuron j in layer $l^{(h)}$. The redistribution rule is applied until it reaches the input layer, which becomes $\mathcal{A}_c(\mathbf{s}_{*,a})$. A modified version of LRP, dubbed LRP-ϵ, is frequently used, which adds a small stabilizer term ϵ in the denominator of Eq (9). The role of ϵ is to absorb some weak or contradictory relevance, thereby leading to sparser and less noisy descriptions [59]. We adopt this LRP-ϵ method in our experiments and use the iNNvestigate toolbox [60] to calculate the attribution maps.

4.2.2 Attribution maps for open set gait recognition. Typically, methods generating attribution maps are applied to neural networks for performing classification tasks [19, 20, 55], which is applicable to the closed set recognition. In this case, each component in the output is interpreted as the inferred probability. In this context, the attribution map $\mathcal{A}_c$ represents the relevance scores of the input to the probability that the input is classified as class c.

However, in the open set recognition setting, the encoder returns embedding vectors for each unit step $\mathbf{s}$, rather than a vector of the probabilities. Due to the fact that the components in the output do not have explicit meaning, it may be hardly possible to directly apply the previous approach for interpretation. In our study, we propose another strategy that averages out the attribution maps over all 128 components in the embedding vectors to see how the trained encoder views the input $\mathbf{s}_{*,a}$. Formally, the attribution map is defined as

$$\mathcal{A}(\mathbf{s}) = \frac{1}{128}\sum_{c=1}^{128}\mathcal{A}_c(\mathbf{s}). \tag{10}$$

5 Evaluation and results

In this section, we first describe the experimental settings and evaluation metrics. Next, we present evaluation results for the proposed gait recognition with the prototyping encoder–decoder architecture. Finally, we comprehensively demonstrate the evaluation results of our XAI methods using attribution maps.

5.1 Experimental settings and evaluation metrics

It is worth noting that one of the key component of the proposed method is utilization of the pressure data. To have pressure data while walking, we collected gait information data by ourselves using the insole from 40 adults as they walked for approximately 3 minutes. The entire dataset consists of 6,303 unit steps, which correspond to 158 unit steps per subject. We also note that, although a much higher number of subjects might be utilized from publicly available datasets (e.g., [6, 7]), we do not adopt such datasets in our experiments since they basically lack pressure sensors that play a crucial role in our study.

To see the impact of our occlusion strategy that replaces the corresponding parts by zero, the original data with pressure sensors of 0's are first replaced by δ through data pre-processing. The δ was set to 0.01 to make minimal adjustments to the original data.

As shown in Fig 7, the data were split into three sets: *training*, *known test*, and *unknown test* sets. First, we sampled 20 subjects from 40 subjects, and all of their unit steps were assigned to the *training* set for the encoder–decoder network. Second, the other 20 subjects were divided into two groups equally. 10 unit steps of 10 subjects in one group were used to train the OSVM classifier, and the remaining unit steps of the subjects in the group were kept for the *known test* set. Finally, 100% of the unit steps of 10 subjects in the other group were allocated to the *unknown test* set. The *known test*, *unknown test*, and *training* sets contain approximately 1,480, 1,580, and 3,160 unit steps, respectively. Such datasets were generated 10 times repeatedly.

We trained and evaluated the network with each dataset. We then evaluated the performance metrics averaged from 10 repetitions. When a unit step in the *known test* set is recognized correctly, we define such an event as a *true positive* (TP), otherwise a *false negative* (FN). In contrast, if a unit step in the *unknown test* set is not recognized as any subject known, we

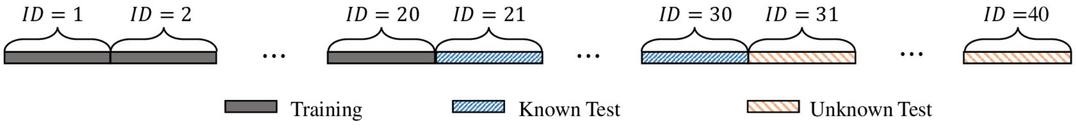

Fig 7. Illustration of splitting the data into the *training, known test*, and *unknown test* sets.

https://doi.org/10.1371/journal.pone.0264783.g007

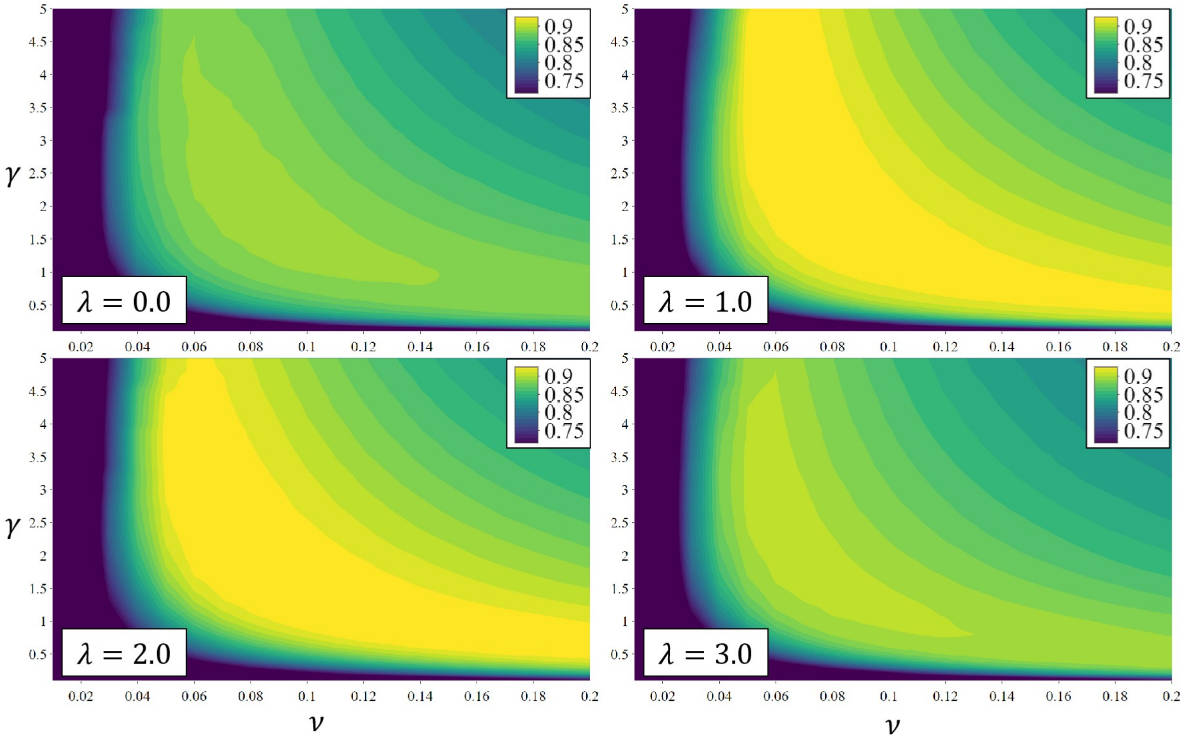

Fig 8. ACC as a function of γ and ν for a value of $\tau = -0.1$. A similar rate is represented as the same color with the maximum 1% difference, with the highest rates as yellow.

https://doi.org/10.1371/journal.pone.0264783.g008

define such an event as a *true negative* (TN) or otherwise a *false positive* (FP). Subsequently, we denote the true positive rate as $TPR = \frac{TP}{TP+FN}$, the true negative rate as $TNR = \frac{TN}{TN+FP}$, and the accuracy as $ACC = \frac{TP+TN}{TP+FN+TN+FP}$.

5.2 Evaluation for gait recognition

The distributions of ACC as a function of hyper-parameters γ and ν for different values of λ are shown in Fig 8. Clearly, selection of γ and ν is critical to the overall recognition accuracy of our model. The area in which the rates are greater than 90% (corresponding to the yellow area) indicates that the region for λ = 1.0 is broader than the regions for other cases. This means that the case of λ = 1.0 has a weak dependency when γ and ν are selected, which affects the robustness to the recognition performance. In practice, the hyper-parameters need to be tuned by considering both the TPR and TNR simultaneously. For example, if all unit steps are rejected, then we could achieve the TNR of 100% at the cost of 0% of TPR. Thus, we set the hyper-parameters in the sense of minimizing the difference between the TPR and TNR.

To examine the effect of τ, we used $\gamma = 2.2$ and $\nu = 0.06$ for λ = 1.0. In Fig 9, we can see that the TPR and ACC get considerably enhanced when τ is smaller than 0. Based on this observation, it is desirable to choose alternative τ instead of $\tau = 0.0$ for the decision boundary in the latent space.

5.3 Evaluation for attribution maps

5.3.1 Extraction of common attribution maps. In our study, we would like to answer the following question: *can we identify what parts of the unit steps are the most relevant to the open*

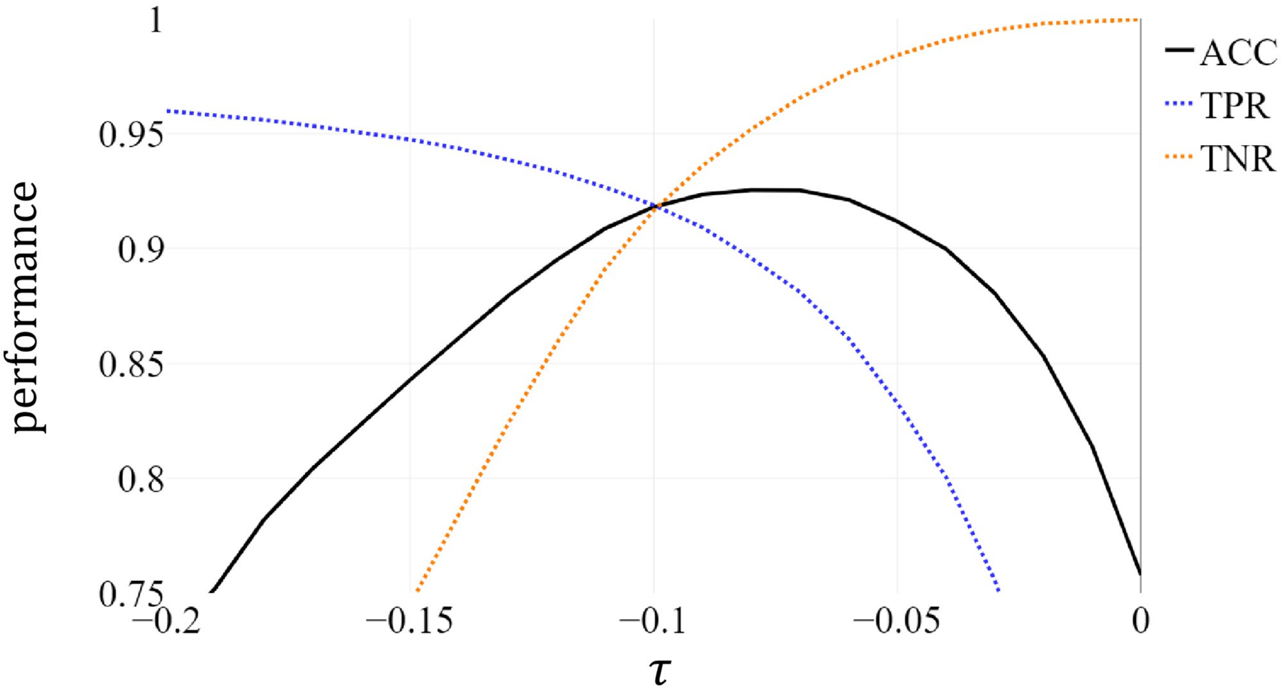

Fig 9. Performance as a function of τ for fixed values of γ = 2.2, ν = 0.06, and λ = 1.0.

https://doi.org/10.1371/journal.pone.0264783.g009

set gait recognition, regardless of the subject id? In other words, we aim to find an attribution map where the relevance values are commonly valid for most of the unit steps. To achieve this goal, we define a *common attribution map* $\mathcal{A}_{com}$ as the attribution map averaged out over all unit steps in the *training* set:

$$\mathcal{A}_{com} = \frac{1}{\sum_a^m n_a} \sum_a^m \sum_i^{n_a} \mathcal{A}(\mathbf{s}_{i,a}), \tag{11}$$

where a denotes the subject index, i denotes the unit step index, m denotes the number of subjects in the *training* set, and n_a denotes the number of unit steps of the subject $id = a$ in the *training* set.

5.3.2 Evaluation methods. As in [54, 55], we choose *region perturbation* to evaluate the common attribution map from Eq (11) with some modifications. For the given unit step $\mathbf{s}$ and the common attribution map $\mathcal{A}_{com}$ (derived from either SA or LRP-ϵ), it is possible to order the relevance scores in $\mathcal{A}_{com}$ from the highest to the lowest. The region perturbation observes the amount of performance degradation when specific parts of the unit step are occluded (i.e., replaced by zero). The intuition behind is that, if we occlude regions of the unit step with high relevance, then the performance degradation will be significant compared to the case in which we occlude regions with low relevance.

For the evaluation, we consider a sequence, denoted as $O = (pos_1, \cdots, pos_i, pos_j, \cdots, pos_L)$, where pos_i indicates the position (e.g., row and column indices) and L is the total number of components in the unit step $\mathbf{s}$. The sequence order is determined by the relevance score of the position, where, for two neighboring indices i and j, the attribution map scores for pos_i are greater than (or equal to) pos_j. Hence, pos_1 indicates the position in the unit step with the highest relevance score, and pos_L indicates the position of the lowest relevance score. As explained in Section 4.2.2, if the output of the network to be explained represents the classification probability, then each output component eventually responds to a binary (yes/no) question. In this case, the sign of the relevance scores can be interpreted as positively/negatively affecting a certain class. However, in our experiment, because each component of the embedding vectors (corresponding to the output of the encoder $f(\cdot)$) has no explicit meaning, we use the *magnitude* of the relevance score to acquire the sequence O.

5.3.3 Evaluation results for common attribution maps. To validate the effectiveness of common attribution maps, we first divide the sequence O into several groups. We equally divide O into five sub-sequences, $O_1, \cdots, O_5$. Thus, O_1 includes the top 20% of the positions with the highest relevance score, and O_2 includes the next 20% of the positions, and so on. We

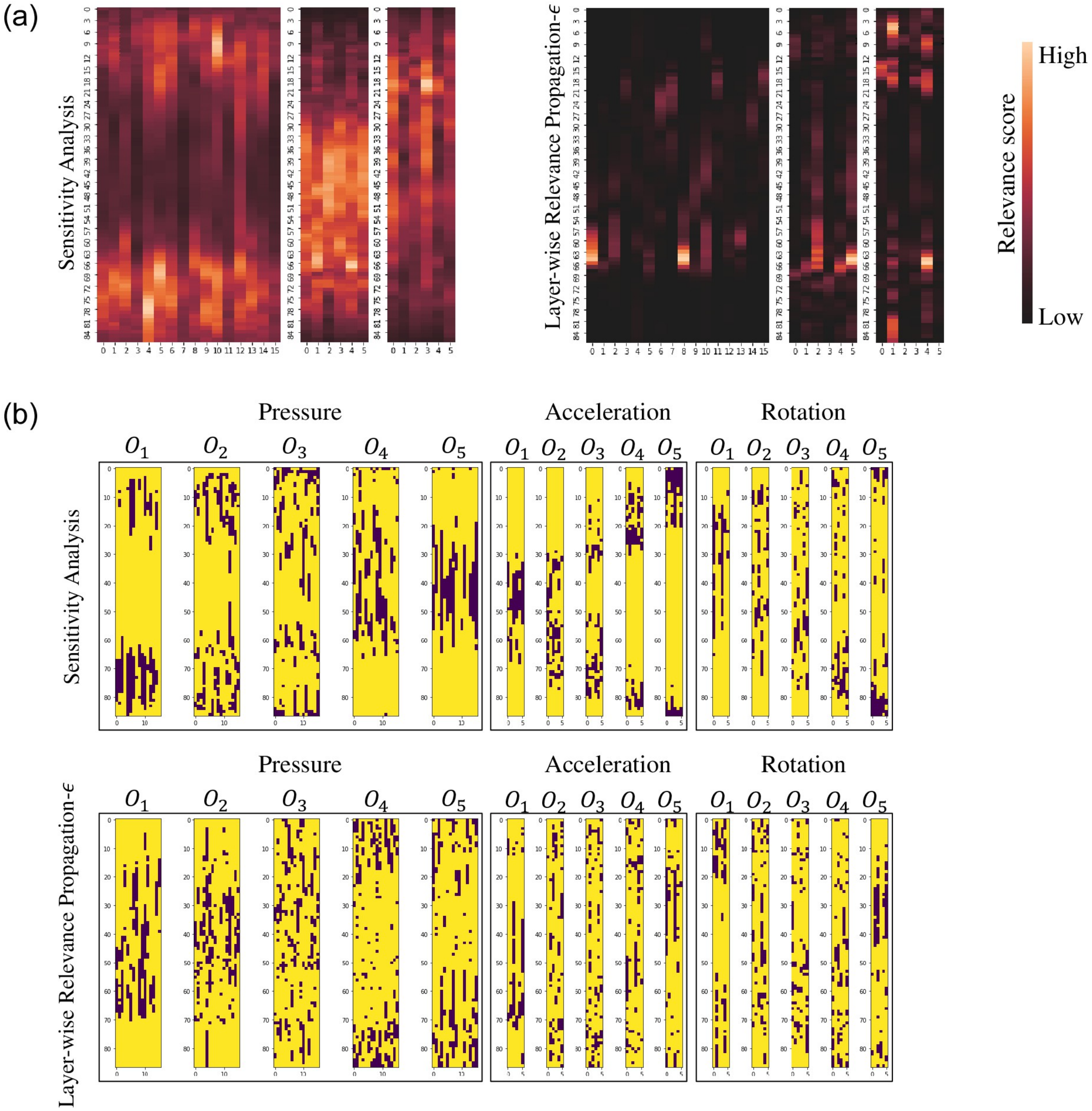

Fig 10. Averaged relevance score heat maps and the their occluding positions for $O_1 \cdots O_5$ of SA and LRP-ϵ. In each heatmap, the x-axis and y-axis indicate features of each sensor and time-stamps of each unit step, respectively. (a) Common attribution maps. (Left: SA, Right: LRP-ϵ). (b) Occluding positions ($O_1, \cdots, O_5$) for all modal inputs. (Top: SA, Bottom: LRP-ϵ).

https://doi.org/10.1371/journal.pone.0264783.g010

removed the positions in the unit step for $O_1, \cdots, O_5$ individually and then observed the metrics ACC, TPR, and TNR. We compare the results with a random baseline such that the same amount is occluded but the positions are randomly chosen.

The heatmap visualizations and their occluding positions are shown in Fig 10a and 10b. As shown in Fig 10a, the attribution maps for both methods exhibit different patterns. For instance, in the pressure attribution maps, LRP-ϵ focuses on the parts of unit steps when feet are contacting with the ground (i.e., the stance phase). In contrast, SA focuses on the other parts of unit steps when feet are in the air (i.e., the swing phase). Subsequently, the occluding positions for $O_1, \cdots, O_5$ also reveal different patterns as depicted in Fig 10b.

Furthermore, from Fig 11a and 11b, we can observe the overall performance degradation for both methods in terms of TPR and ACC as we move from occlusion of O_5 to that of O_1. The occlusion of O_1 obtained from the attribution map results in TPR decrement of 0.3 and 0.5 for SA and LRP-ϵ, respectively, compared to the random baseline. This implies that both SA and LRP-ϵ can detect the most important unit step regions (O_1) for the *known test* set. Especially, we observe that, if we occlude O_1, then the TPR drops to zero for LRP-ϵ. When we pay our attention to less relevant occlusions, we see that the SA method exhibits a lower performance gain in TPR, resulting in inferior performance compared to the random baseline even in the case of O_5. Meanwhile, LRP-ϵ surpasses the random baseline in O_3 and results in much superior performance in O_5. This demonstrates that using LRP-ϵ can offer more robust common attribution maps compared to SA. For both methods for O_1, the TNR values are very close to 1 and the TPR values are very low. From these findings, we see that the model does not recognize almost every unit step. In consequence, both methods detect the most relevant part of the unit step while LRP-ϵ outperforms SA as seen from the ACC.

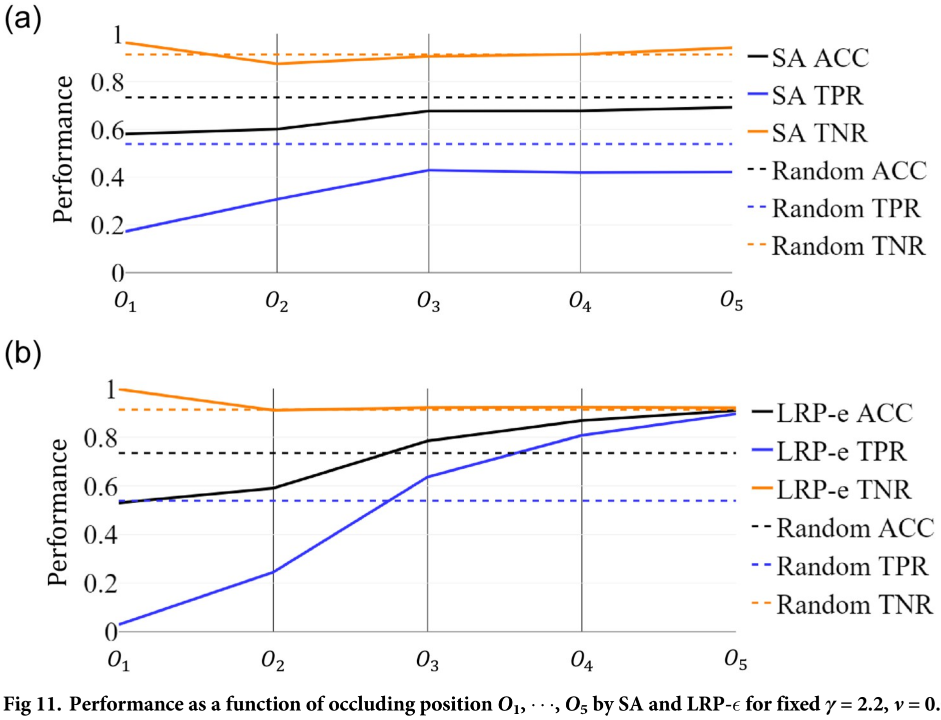

Fig 11. Performance as a function of occluding position $O_1, \cdots, O_5$ by SA and LRP-ϵ for fixed $\gamma = 2.2$, $\nu = 0.06$, $\tau = -0.1$, and $\lambda = 1.0$. (a) SA. (b) LRP-ϵ.

https://doi.org/10.1371/journal.pone.0264783.g011

6 Concluding remarks

This paper presented a novel gait recognition system capable of revealing the most important parts of the multimodal time series to distinguish the individuals. The proposed encoder–decoder prototyping network architecture along with our loss functions successfully solved the open set gait recognition problem from the data collected using a wearable device. Our experiments demonstrated that the system's recognition performance is less sensitive to the changes in the values of hyper-parameters than those in the previous studies. Furthermore, using XAI methods based on SA and LPR-ϵ, we provided insightful interpretability for the complex relations between the multimodal time series and the recognition results. The proposed common attribution map clearly revealed which part of the multimodal time series is relevant to the recognition performance.

Potential avenues of future research in this area include performance improvement of the end-to-end recognition system and design of more sophisticated XAI methods by optimizing the common attribution map.

Author Contributions

Conceptualization: Jucheol Moon, Won-Yong Shin, Sang-Il Choi.

Data curation: Jin-Duk Park, Nelson Hebert Minaya.

Formal analysis: Jucheol Moon, Won-Yong Shin, Sang-Il Choi.

Funding acquisition: Won-Yong Shin, Sang-Il Choi.

Investigation: Jucheol Moon, Jin-Duk Park, Won-Yong Shin, Sang-Il Choi.

Methodology: Jucheol Moon, Yong-Min Shin, Won-Yong Shin, Sang-Il Choi.

Project administration: Jucheol Moon, Won-Yong Shin, Sang-Il Choi.

Software: Yong-Min Shin, Jin-Duk Park, Nelson Hebert Minaya.

Supervision: Jucheol Moon, Won-Yong Shin, Sang-Il Choi.

Validation: Jucheol Moon, Yong-Min Shin, Jin-Duk Park, Nelson Hebert Minaya, Won-Yong Shin, Sang-Il Choi.

Visualization: Yong-Min Shin, Jin-Duk Park.

Writing – original draft: Jucheol Moon.

Writing – review & editing: Jucheol Moon, Yong-Min Shin, Jin-Duk Park, Nelson Hebert Minaya, Won-Yong Shin, Sang-Il Choi.

References

1. Wahid F, Begg RK, Hass CJ, Halgamuge S, Ackland DC. Classification of Parkinson's disease gait using spatial-temporal gait features. IEEE J Biomed Health Inform. 2015; 19(6):1794–1802. https://doi.org/10.1109/JBHI.2015.2450232 PMID: 26551989
2. Liao R, Yu S, An W, Huang Y. A model-based gait recognition method with body pose and human prior knowledge. Pattern Recognit. 2020; 98:107069. https://doi.org/10.1016/j.patcog.2019.107069
3. Wan C, Wang L, Phoha VV. A survey on gait recognition. ACM Comput Surv. 2018; 51(5):1–35. https://doi.org/10.1145/3230633
4. Scheirer WJ, de Rezende Rocha A, Sapkota A, Boult TE. Toward open set recognition. IEEE Trans Pattern Anal Mach Intell. 2012; 35(7):1757–1772. https://doi.org/10.1109/TPAMI.2012.256
5. Geng C, Huang S, Chen S. Recent Advances in Open Set Recognition: A Survey. IEEE Trans Pattern Anal Mach Intell. 2021; 43(10):3614–3631. https://doi.org/10.1109/TPAMI.2020.2981604 PMID: 32191881

6. Ngo TT, Makihara Y, Nagahara H, Mukaigawa Y, Yagi Y. The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication. Pattern Recognit. 2014; 47 (1):228–237. https://doi.org/10.1016/j.patcog.2013.06.028

7. Al Kork SK, Gowthami I, Savatier X, Beyrouthy T, Korbane JA, Roshdi S. Biometric database for human gait recognition using wearable sensors and a smartphone. In: Proc. Int. Conf. Bio-engineering for Smart Technologies (BioSMART). Paris, France; 2017. p. 1–4.

8. Moon J, Minaya NH, Le NA, Park HC, Choi SI. Can Ensemble Deep Learning Identify People by Their Gait Using Data Collected from Multi-Modal Sensors in Their Insole? Sensors. 2020; 20(14):4001. https://doi.org/10.3390/s20144001 PMID: 32708442

9. Singh JP, Jain S, Arora S, Singh UP. Vision-based gait recognition: A survey. IEEE Access. 2018; 6:70497–70527. https://doi.org/10.1109/ACCESS.2018.2879896

10. Lee SS, Choi ST, Choi SI. Classification of Gait Type Based on Deep Learning Using Various Sensors with Smart Insole. Sensors. 2019; 19(8):1757. https://doi.org/10.3390/s19081757 PMID: 31013773

11. Schroff F, Kalenichenko D, Philbin J. FaceNet: A unified embedding for face recognition and clustering. In: Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR). Boston, MA; 2015. p. 815–823.

12. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res. 2010; 11 (110):3371–3408.

13. Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC. Support vector method for novelty detection. In: Proc. Adv. Neural Inf. Process. Syst. (NIPS). Denver, CO; 1999. p. 582–588.

14. Samek W, Müller K. Towards Explainable Artificial Intelligence. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. vol. 11700 of Lecture Notes Comput. Sci.; 2019. p. 5–22.

15. Montavon G, Lapuschkin S, Binder A, Samek W, Müller KR. Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recognit. 2017; 65:211–222. https://doi.org/10.1016/j.patcog.2016.11.008

16. Yeom SK, Seegerer P, Lapuschkin S, Binder A, Wiedemann S, Müller KR, et al. Pruning by explaining: A novel criterion for deep neural network pruning. Pattern Recognit. 2021; 115:107899. https://doi.org/10.1016/j.patcog.2021.107899

17. Dindorf C, Teufl W, Taetz B, Bleser G, Fröhlich M. Interpretability of Input Representations for Gait Classification in Patients after Total Hip Arthroplasty. Sensors. 2020; 20(16):4385. https://doi.org/10.3390/s20164385 PMID: 32781583

18. Horst F, Slijepcevic D, Lapuschkin S, Raberger AM, Zeppelzauer M, Samek W, et al. On the Understanding and Interpretation of Machine Learning Predictions in Clinical Gait Analysis Using Explainable Artificial Intelligence. arXiv preprint arXiv:191207737. 2019.

19. Montavon G, Samek W, Müller KR. Methods for interpreting and understanding deep neural networks. Digit Signal Process. 2018; 73:1–15. https://doi.org/10.1016/j.dsp.2017.10.011

20. Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS One. 2015; 10(7):e0130140. https://doi.org/10.1371/journal.pone.0130140 PMID: 26161953

21. Niyogi SA, Adelson EH, et al. Analyzing and recognizing walking figures in XYT. In: Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR). vol. 94. Seattle, WA; 1994. p. 469–474.

22. Zhang Z, Tran L, Yin X, Atoum Y, Liu X, Wan J, et al. Gait Recognition via Disentangled Representation Learning. In: Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR). Long Beach, CA; 2019. p. 4710–4719.

23. Li C, Min X, Sun S, Lin W, Tang Z. DeepGait: A learning deep convolutional representation for view-invariant gait recognition using joint Bayesian. Appl Sciences. 2017; 7(3):210. https://doi.org/10.3390/app7030210

24. Chen X, Xu J, Weng J. Multi-gait recognition using hypergraph partition. Mach Vision Appl. 2017; 28(1-2):117–127. https://doi.org/10.1007/s00138-016-0810-6

25. Wu Z, Huang Y, Wang L, Wang X, Tan T. A comprehensive study on cross-view gait based human identification with deep CNNs. IEEE Trans Pattern Anal Mach Intell. 2016; 39(2):209–226. https://doi.org/10.1109/TPAMI.2016.2545669 PMID: 27019478

26. Tian Y, Wei L, Lu S, Huang T. Free-view gait recognition. PloS One. 2019; 14(4):e0214389. https://doi.org/10.1371/journal.pone.0214389 PMID: 30990804

27. Dehzangi O, Taherisadr M, ChangalVala R. IMU-based gait recognition using convolutional neural networks and multi-sensor fusion. Sensors. 2017; 17(12):2735. https://doi.org/10.3390/s17122735 PMID: 29186887

28. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proc. Adv. Neural Inf. Process. Syst. (NIPS). Lake Tahoe, NV; 2012. p. 1097–1105.

29. Moufawad eAC, Lenoble-Hoskovec C, Paraschiv-Ionescu A, Major K, Büla C, Aminian K. Instrumented shoes for activity classification in the elderly. Gait & Posture. 2016; 44:12–17. https://doi.org/10.1016/j.gaitpost.2015.10.016 PMID: 27004626

30. Gadaleta M, Rossi M. IDNet: Smartphone-based gait recognition with convolutional neural networks. Pattern Recognit. 2018; 74:25–37. https://doi.org/10.1016/j.patcog.2017.09.005

31. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995; 20(3):273–297. https://doi.org/10.1007/BF00994018

32. Choi SI, Moon J, Park HC, Choi ST. User Identification from Gait Analysis Using Multi-Modal Sensors in Smart Insole. Sensors. 2019; 19(17):3785. https://doi.org/10.3390/s19173785 PMID: 31480467

33. Luo Y, Coppola SM, Dixon PC, Li S, Dennerlein JT, Hu B. A database of human gait performance on irregular and uneven surfaces collected by wearable sensors. Scientific Data. 2020; 7(1):219. https://doi.org/10.1038/s41597-020-0563-y PMID: 32641740

34. Weiss GM, Yoneda K, Hayajneh T. Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. IEEE Access. 2019; 7:133190–133202. https://doi.org/10.1109/ACCESS.2019.2940729

35. Al Kork SK, Gowthami I, Savatier X, Beyrouthy T, Korbane JA, Roshdi S. Biometric database for human gait recognition using wearable sensors and a smartphone. In: Proc. Int. Conf. Bio-engineering Smart Technol. (BioSMART); 2017. p. 1–4.

36. Chereshnev R, Kertész-Farkas A. HuGaDB: Human Gait Database for Activity Recognition from Wearable Inertial Sensor Networks. In: Proc. Analy. Images, Social Netw. Texts (AIST). vol. 10716. Moscow, Russia; 2017. p. 131–141.

37. Subramanian R, Sarkar S, Labrador MA, Contino K, Eggert C, Javed O, et al. Orientation invariant gait matching algorithm based on the Kabsch alignment. In: Int. Conf. Identity, Security Behavior Anal. (ISBA). Hong Kong, China; 2015. p. 1–8.

38. Thanh Trung N, Makihara Y, Nagahara H, Mukaigawa Y, Yagi Y. Orientation-Compensative Signal Registration for Owner Authentication Using an Accelerometer. IEICE Trans Inf Syst. 2014; 97:541–553.

39. Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL. A Public Domain Dataset for Human Activity Recognition using Smartphones. In: Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN); 2013.

40. Frank J, Mannor S, Pineau J, Precup D. Time Series Analysis Using Geometric Template Matching. IEEE Trans Pattern Anal Mach Intell. 2013; 35(3):740–754. https://doi.org/10.1109/TPAMI.2012.121 PMID: 22641699

41. Reiss A, Stricker D. Introducing a New Benchmarked Dataset for Activity Monitoring. In: Proc. 16th Int. Symp. Wearable Computers (ISWC). Newcastle, United Kingdom; 2012. p. 108–109.

42. Zhang M, Sawchuk AA. USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In: Dey AK, Chu H, Hayes GR, editors. Proc. 2012 ACM Conf. Ubiquitous Comput. (Ubicomp). Pittsburgh, PA; 2012. p. 1036–1043.

43. Comparative study on classifying human activities with miniature inertial and magnetic sensors. Pattern Recognit. 2010; 43(10):3605–3620. https://doi.org/10.1016/j.patcog.2010.04.019

44. Bächlin M, Plotnik M, Roggen D, Maidan I, Hausdorff JM, Giladi N, et al. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. IEEE Trans Inf Technol Biomed. 2010; 14(2):436–446. https://doi.org/10.1109/TITB.2009.2036165 PMID: 19906597

45. Gafurov D, Snekkenes E, Bours P. Improved Gait Recognition Performance Using Cycle Matching. In: Workshops Int. Conf. Adv. Inf. Netw. Appl. (WAINA). Perth, Australia; 2010. p. 836–841.

46. Alotaibi M, Mahmood A. Improved gait recognition based on specialized deep convolutional neural network. Comput Vision Image Understanding. 2017; 164:103–110. https://doi.org/10.1016/j.cviu.2017.10.004

47. Moon J, Le NA, Minaya NH, Choi SI. Multimodal Few-Shot Learning for Gait Recognition. Appl Sciences. 2020; 10(21):7619. https://doi.org/10.3390/app10217619

48. Chen HY, Lee CH. Vibration signals analysis by explainable artificial intelligence (XAI) approach: Application on bearing faults diagnosis. IEEE Access. 2020; 8:134246–134256. https://doi.org/10.1109/ACCESS.2020.3006491

49. Kuzlu M, Cali U, Sharma V, Güler Ö. Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. IEEE Access. 2020; 8:187814–187823. https://doi.org/10.1109/ACCESS.2020.3031477

50. Adadi A, Berrada M. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). Inst Elect Electronics Engineers Access. 2018; 6:52138–52160.

51. Sundararajan M, Taly A, Yan Q. Axiomatic Attribution for Deep Networks. In: Proc. Int. Conf. Mach. Learn. (ICML). Sydney, Australia; 2017. p. 3319–3328.

52. Shrikumar A, Greenside P, Kundaje A. Learning Important Features Through Propagating Activation Differences. In: Proc. Int. Conf. Mach. Learn. (ICML). Sydney, Australia; 2017. p. 3145–3153.

53. Arras L, Horn F, Montavon G, Müller KR, Samek W. "What is relevant in a text document?": An interpretable machine learning approach. PloS One. 2017; 12(8):e0181142. https://doi.org/10.1371/journal.pone.0181142 PMID: 28800619

54. Samek W, Binder A, Montavon G, Lapuschkin S, Müller K. Evaluating the Visualization of What a Deep Neural Network Has Learned. IEEE Trans Neural Networks Learn Systems. 2017; 28(11):2660–2673. https://doi.org/10.1109/TNNLS.2016.2599820 PMID: 27576267

55. Ancona M, Ceolini E, Öztireli C, Gross M. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In: Proc. Int. Conf. Learn. Representations (ICLR). Vancouver, Canada; 2018.

56. 3LLabs. Footlogger Insole;. http://footlogger.com/hp_new/?page_id=11.

57. Wen Y, Zhang K, Li Z, Qiao Y. A discriminative feature learning approach for deep face recognition. In: Proc. Eur. Conf. Comput. Vision (ECCV). Amsterdam, The Netherlands; 2016. p. 499–515.

58. Kim J, Oh TH, Lee S, Pan F, Kweon IS. Variational prototyping-encoder: One-shot learning with prototypical images. In: Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR). Long Beach, CA; 2019. p. 9462–9470.

59. Montavon G, Binder A, Lapuschkin S, Samek W, Müller KR. Layer-wise relevance propagation: An overview. In: Explainable AI: Interpreting, Explaining Visualizing Deep Learn.; 2019. p. 193–209.

60. Alber M, Lapuschkin S, Seegerer P, Hägele M, Schütt KT, Montavon G, et al. iNNvestigate Neural Networks! J Mach Learn Research. 2019; 20(93):1–8.