Multimodal Few-Shot Learning for Gait Recognition

Jucheol Moon 1*, Nhat Anh Le 1, Nelson Hebert Minaya 1 and Sang-Il Choi 2,*

1 Department of Computer Engineering and Computer Science, California State University, Long Beach, CA 90840, USA; jucheol.moon@csulb.edu (J.M.); Nhat.Le01@student.csulb.edu (N.A.L.); Nelson.Minaya@student.csulb.edu (N.H.M.)
2 Department of Computer Science and Engineering, Dankook University, Yongin-si 16890, Gyeonggi-do, Korea
* Correspondence: choisi@dankook.ac.kr

Received: 8 October 2020; Accepted: 25 October 2020; Published: 29 October 2020

Abstract: A person’s gait is a behavioral trait that is uniquely associated with each individual and can be used to recognize the person. As information about the human gait can be captured by wearable devices, a few studies have led to the proposal of methods to process gait information for identification purposes. Despite recent advances in gait recognition, an open set gait recognition problem presents challenges to current approaches. To address the open set gait recognition problem, a system should be able to deal with unseen subjects who have not included in the training dataset. In this paper, we propose a system that learns a mapping from a multimodal time series collected using insole to a latent (embedding vector) space to address the open set gait recognition problem. The distance between two embedding vectors in the latent space corresponds to the similarity between two multimodal time series. Using the characteristics of the human gait pattern, multimodal time series are sliced into unit steps. The system maps unit steps to embedding vectors using an ensemble consisting of a convolutional neural network and a recurrent neural network. To recognize each individual, the system learns a decision function using a one-class support vector machine from a few embedding vectors of the person in the latent space, then the system determines whether an unknown unit step is recognized as belonging to a known individual. Our experiments demonstrate that the proposed framework recognizes individuals with high accuracy regardless they have been registered or not. If we could have an environment in which all people would be wearing the insole, the framework would be used for user verification widely.

Keywords: gait analysis; open set recognition; few-shot learning; multi-modality; wearable sensors

1. Introduction

The human gait, a person’s manner of walking, is sufficient for discriminating between individuals [1–3]. Information about a person’s gait has been utilized for diagnosing diseases [4–7], and it can also be used for biometric authentication [8–13]. Gait recognition has three main advantages compared to other typical biometric authentication methods. First, it is robust against impersonation attacks. Second, it does not require physical contact between sensors and people. Lastly, it may not be necessary to use vision sensors to capture gait information [14,15].

A typical framework for gait recognition consists of two parts: capturing data that are a good representation of the gait, and using algorithms to classify the collected data to identify individuals. In that sense, we can categorize the gait recognition framework based on data acquisition devices and data analysis algorithms. Specifically, information about the gait can be collected using vision sensors, pressure sensors, and inertial measurement units (IMUs), then the collected data can be analyzed using linear discriminant analysis (LDA), k-nearest neighbor (k-NN), hidden Markov model (HMM), support vector machine (SVM), convolutional neural network (CNN), or combinations thereof [15].
In general, two types of recognition problems exist. The first one is closed set recognition, whereby all testing classes are known at the time of training, and the other one is open set recognition where incomplete knowledge is given at the time of training, and unknown classes can be classified by an algorithm during testing [16]. In gait recognition, the majority of frameworks are designed to solve the closed set recognition problem, and few approaches attempt to address the open set recognition problem [17].

In a recent study of the closed set recognition problem, the original data were divided into separate unit steps to use the data more efficiently and effectively [13]. In this study, we adapted their method to recognize individuals from their gait. The pressure sensors, 3D-axis accelerometer and 3D-axis gyroscope installed in the insoles of shoes record the time series data of gait information [18]. Human walking cycles consist of a stance phase and a swing phase [19]. During the swing phase, since the entire foot is in the air, the values reported by the pressure sensors should be zero. Considering that, we might be able to divide the original time-series data into consecutive unit steps by detecting the time index where the pressure value is zero. However, due to the inference between sensors or high temperatures in the insole, the reported pressure values are often non-zero during the swing phase [20]. To avoid potential errors, these authors determined the unit steps using Gaussian smoothing [13].

To recognize individuals from consecutive unit step data, we propose the use of an ensemble consisting of a CNN and a recurrent neural network (RNN). The ensemble network maps multimodal unit step data to the embedding vectors in a latent space. To evaluate the system, we use training, unknown known, unknown unknown datasets. In the training phase, the system is provided all labeled samples in training dataset, a few (3 ≤ k ≤ 10) labeled samples in unknown known dataset. To train this network, we used triplet loss [21], which forces the distances between the embedding vectors of the homogeneous unit steps to be much smaller than the distances between the embedding vectors of the heterogeneous unit steps in training dataset. Once the ensemble is trained, we randomly select k unit steps for every person in the unknown known dataset, and we store the corresponding k embedding vectors and the centroid of the embedding vectors. Using the one-class support vector machine (OSVM) algorithm [22], we compute a decision function for the k embedding vectors of each individual who was included in the unknown known dataset.

In the test phase, a unit step in unknown known dataset (except selected k in the training phase) or unknown unknown dataset is given, in response to which the system should indicate whether the given unit step belongs to someone who was included in the unknown known dataset. The system accomplishes this by mapping the unknown unit step to an embedding vector using the ensemble network and finds the nearest neighbor by simply using the distances between the embedding vector of an unknown unit step and the centroids of individuals in the unknown known dataset. Finally, we conclude that the unknown unit step belongs to the nearest neighbor if the embedding vector of the unknown unit step is inside the decision boundary of the nearest neighbor.

In summary, the contributions of this study are as follows: (1) We designed an ensemble network that uses CNN and RNN, which is applicable to open set gait recognition. (2) We developed a system that addresses the open set gait recognition problem using the OSVM algorithm. (3) The system requires only a few walking cycles of an individual to be able to recognize them.

**Related Work**

Studies on gait recognition began with the use of vision sensors [23]. These approaches were subsequently further studied and developed [24–29]. In general, vision-based gait recognition requires strict conditions while collecting the data. For example, a video sequence would have to contain only individuals that need to be recognized. Apart from this, the recognition accuracy is not sufficiently high. Moreover, the sensing devices’ viewpoint and the orientation also affect the accuracy. To recognize a subject from a video sequence which includes more than one person, each subject should be segmented and tracked individually [30,31]. To achieve a stable recognition accuracy regardless
sensing devices’ viewpoint and orientation, 3D construction model or view transformation model can be utilized [32–34].

In recent studies, pressure sensors and IMUs are widely used to collect data. Typically, IMUs consist of an accelerometer, a gyroscope, and a magnetometer. For instance, gait information was collected from IMUs placed on the chest, lower back, right wrist, knee, and ankle of subjects [8], and then a CNN-based predictive model [35] identified individuals. Similarly, gait information was collected from a variety of IMUs attached to the user in multiple positions, and the user’s activity was recognized by analyzing the time series patterns of the data [36]. Later, pressure sensors and IMUs were installed in wearable devices, such as, fitness trackers, smartphones, or shoe insoles [37]. For example, gait information was collected using IMUs installed in smartphones [38]. Subjects carried the smartphones in their front trouser pockets to gather data, and then a mixed model consisting of CNN and SVM [39] was used to recognize individuals. In another study, gait information was measured using pressure sensors and an accelerometer on the shoe insoles [40], and the collected data were classified using null space LDA [41]. However, these methods require the placement of different types of sensors on various parts of the body, take a long period of time to gather data, or need improvement in terms of identification accuracy. More recently, an ensemble network was used to identify individuals using gait information, but their framework is only effective for solving the closed set recognition problems [13].

The open set gait recognition problem was partially addressed in the literature. For example, gait information is captured by 11 cameras, and it was classified using CNN with softmax output layer [42]. To address the open set gait recognition problem, the softmax layer included one more class than the number of subjects in the training dataset. To train the network, samples of subjects who were not included in the training dataset are labeled as ‘not recognized.’ Because of using the softmax output layer, this approach is not scalable since the network should be trained again every time a new subject is added. For another example, collected gait information using IMUs installed in smartphones was recognized using a framework based on CNN and OSVM [38]. Different than our study, the proposed method required about a hundred unit steps to train the OSVM algorithm, and the system was evaluated using unit steps in the unknown dataset only.

2. Method

In our work, subjects’ gait information was measured using a shoe insole. The original data format is a vector of time series that consists of consecutive unit steps. We processed the time series vector into fixed size fragments (i.e., unit steps) to improve the recognition accuracy and reduce the computational complexity. These unit steps are then recognized using the proposed system.

2.1. Data Pre-Processing

We used a commercial shoe insole, FootLogger [43], to record subjects’ gait information. The design of the insole is depicted in Figure 1. The insole for each foot has eight pressure sensors, a 3D-axis accelerometer, and a 3D-axis gyroscope. The pressure sensor measures the level of pressure at one of three levels: 0, 1, or 2. The accelerometer and gyroscope gauge acceleration and rotation in three dimensions as integers between −32,768 and 32,768. The sampling rate of the insole was 100 Hz, and we collected data from both of the subjects’ feet.
Figure 1. Design of the shoe insole used in our experiments.

We followed the notation of a previous study [13]. We denote the (univariate) time series and multivariate time series by \( x(t) \) and \( x(t) \), respectively. Different sensing modalities are expressed using superscript letters, that is \( x^p(t) \) for pressure, \( x^a(t) \) for acceleration, and \( x^r(t) \) for rotation. Different subject identifications are expressed using subscript numbers, that is \( x_{i}(t) \) for \( id = i \). We also adapt the method from Reference [13] to determine the unit steps from the original time series. Except, for brevity, we use the notation \( s \) and \( s(t) \) interchangeably. We repeat the notation here for the readers: The \( i \)th unit step of subject \( id = a \) for sensing modality \( m \) is denoted by \( s^m_{i,a} \), and their dimensions are \( |s_{i,a}^{prs}| = d \times (8 \cdot 2) \) and \( |s_{i,a}^{acc}| = |s_{i,a}^{rot}| = d \times (3 \cdot 2) \). Examining the minimum length of the subjects’ unit steps, we set \( d = 87 \) in the experiments. Using the timestamps of the unit steps, the original time series of both feet were converted into the standard format. The procedure of converting to the standard format also follows Reference [13]. We omit the detailed information to avoid repetition.

2.2. Network Architecture

We adapted and upgraded the networks that were used previously [13]. Figure 2 depicts the design of the network architecture. A number of different architectures (including shallower/deeper and narrower/wider) were evaluated and compared, however, their performance differences were neglectable. The original datasets include time series data of pressure, acceleration and rotation. Since the pressure sensors measured the pressure on the same foot during the same walking cycle, we assumed that their values are correlated. Similar assumptions were made for three-dimensional acceleration and rotation values. By considering these correlations, we designed a encoding network model combining CNN and RNN. The proposed network model maps unit steps of pressure \( s^{prs} \), acceleration \( s^{acc} \), and rotation \( s^{rot} \) in the standard format to embedding vectors \( v \):

\[
f(s^{prs}, s^{acc}, s^{rot}) = v.\tag{1}
\]

We use the notation \( f_{cnn} \) and \( v^{cnn} \) when only the CNN is activated, \( f_{rnn} \) and \( v^{rnn} \) when only the RNN is activated, and \( f_{ens} \) and \( v^{ens} \) when both CNN and RNN are activated.
2.3. Convolutional Neural Network

Given each sensing mode, our proposed CNN includes three identical networks that function independently, and the outputs of these three networks are concatenated. Each network contains three one-dimensional (1D) convolutional layers with 32, 64, and 128 filters, and the convolutional layers are followed by a batch normalization layer. Each filter in the first convolutional layer has a size of $20 \times (w \cdot 2)$, while the sizes of filters are $20 \times 32$ and $20 \times 64$ for the second and third convolutional layers, respectively. More specifically, for the first convolutional layer, the width of each filter is equal to that of the standard format ($w \cdot 2$). We slide each filter across the height of the input and compute the dot product between the filter and the input, resulting in a series of scalar values. This convolution operation is repeated for the 32 filters, and the resulting series of scalar values are stacked horizontally, whereby the width of the output becomes equal to the number of filters. The stride of all convolutional layers is set to 1, and the padding size is set such that the height of the output is the same as that of the input. Similarly, for the second and third convolutional layers, the width of the filters equals the number of filters in the previous convolutional layer. Therefore, the shape of the feature map is $87 \times 32$, $87 \times 64$, and $87 \times 128$ after each convolutional layer. The last feature map is flattened, then the size of the feature vector is $87 \cdot 128$. The feature vectors of the three networks are concatenated to form one vector, followed by two fully connected layers. We use a rectifier linear unit (ReLU) as the activation function for every convolutional layer and the first fully connected layer to avoid the vanishing gradient phenomenon [44].

2.4. Recurrent Neural Network

Similarly, our proposed RNN includes three identical networks that operate independently, given each sensing mode. The outputs of these three networks are ultimately concatenated. Each network contains two consecutive long short-term memory (LSTM) layers [45]. LSTM is a modified version of RNN with the capability of utilizing internal memory units to overcome the vanishing gradient problem of traditional RNN models. More specifically, we include 128 memory units in each LSTM layer and activate the input, output, and forget gate by the sigmoid function.
To prevent overfitting, the dropout rate was set at 0.2 \[46\]. Similar to the CNN, the input for the first LSTM layer has the shape 87 \(\times\) \((w \cdot 2)\). For each row of input data, the LSTM layer creates a scalar value per memory unit; the resulting scalar values are concatenated to form an output vector of shape 87 \(\times\) 128. The second LSTM layer returns the scalar value per memory unit, therefore, the size of the output vector is 128. The output vectors of the three networks are concatenated to form one vector, followed by two fully connected layers.

### 2.5. Embedding Vector

We take the last fully connected layer with 128 units of the CNN and RNN as the output of each network model. Therefore, the dimensions of the embedding vectors of CNN and RNN are identical to 128, that is \(f_{\text{cnn}}(\cdot) \in \mathbb{R}^{128}, f_{\text{rnn}}(\cdot) \in \mathbb{R}^{128}\). The embedding vector of the ensemble model is generated by concatenating the embedding vectors of CNN and RNN; hence, the dimension is 256, that is \(f_{\text{ens}}(\cdot) \in \mathbb{R}^{256}\).

All embedding vectors are normalized, that is \(||f_{\text{cnn}}(\cdot)||_2 = ||f_{\text{rnn}}(\cdot)||_2 = ||f_{\text{ens}}(\cdot)||_2 = 1||\)

### 2.6. Loss Function

Let \(s_{i,a}^m\) and \(s_{j,a}^m\) \((i \neq j)\) be two unit steps of the subject \(id = a\) for a sensing modality \(m\), and let \(s_{k,b}^m\) be a unit step of subject \(id = b\) for a sensing modality \(m\). The model takes three types of unit steps: pressure, acceleration, and rotation. For brevity, however, we use the simplified notation \(f(s_{i,a})\) instead of \(f(s_{i,a}^p, s_{i,a}^a, s_{i,a}^r)\). Similar to the triplet loss \[21\], the multimodal triplet loss is defined as

\[
L = ||v_{i,a} - v_{j,a}||_2^2 - ||v_{i,a} - v_{k,b}||_2^2 + \alpha,
\]

where \(v_{i,a} = f(s_{i,a}), v_{j,a} = f(s_{j,a}), v_{k,b} = f(s_{k,b})\), and \(\alpha\) is a margin (we set \(\alpha = 1.0\)). The multimodal triplet loss forces that the distance between \(v_{i,a}\) and \(v_{j,a}\) is smaller than the distance between \(v_{i,a}\) and \(v_{k,b}\) for all possible triplets in the training dataset. A conceptual diagram of the multimodal loss is illustrated in Figure 3.

**Figure 3. Illustration of the multimodal triplet loss.**

### 2.7. Few-Shot Learning

We define the unknown known and unknown unknown datasets. In the unknown known dataset, the samples (unit steps) are not used for training the encoding function (i.e., the CNN, RNN, or ensemble networks); instead, only a few samples are utilized for training the decision boundaries of individuals using OSVM. In the unknown unknown dataset, the samples are used only for testing.

For a positive integer \(3 \leq n \leq 10\), let \(\{s_{i,a}\}_{1 \leq i \leq n}\) be the set of randomly selected unit steps of subject \(id = a\) in the unknown known dataset and \(\{v_{i,a} = f(s_{i,a})\}_{1 \leq i \leq n}\) be the set of corresponding embedding vectors that are generated by the trained network model which can be one of CNN, RNN, or ensemble network. For each subject in the unknown known dataset, at first, the system computes the centroid of \(n\) embedding vectors. The centroid of the subject \(id = a\) is defined by \(M_a = \frac{1}{n} \sum_{i=1}^{n} v_{i,a}\). In addition, the system learns decision functions in the latent space using the
OSVM algorithm [22] for all subjects. The algorithm obtains \{v_i, a|1 \leq i \leq n\} as an input and solves the following optimization problem:

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(v_i, a, v_j, a) \\
\text{subject to:} & \quad 0 \leq \alpha_i \leq \frac{1}{\nu n}, \sum_{i=1}^{n} \alpha_i = 1,
\end{align*}
\]

(3)

where \(K(v, v') = e^{-\gamma ||v-v'||^2}\) is a radial bias kernel function, \(\alpha_i\) are the Lagrange multipliers, and \(\gamma\) and \(\nu\) are among the hyper-parameters of the system.

Let \(s_{*,u}\) be a unit step of an unknown subject \(u\) in either the unknown known or unknown unknown dataset. The symbol * denotes that the unit step can be any unit step of the subject \(u\). For each subject \(a\) in the unknown known dataset, the decision function of \(v_{*,u}\) for subject \(id = a\) is defined by \(h_a(v_{*,u}) = \sum_i^a \alpha_i K(v_{i,a}, v_{*,u}) - \rho_a\), where \(\rho_a = \sum_i^a \alpha_i K(v_{i,a}, v_{h,a})\) for any \(h\) that satisfies the condition \(0 < \alpha_h < \frac{1}{\nu n}\) and \(1 \leq h \leq n\). An unknown subject could be one who was included in either the unknown known dataset or unknown unknown dataset. Therefore, the system should be able to recognize a unit step if it belongs to a subject in the unknown known dataset. On the other hand, the system should be able to reject a unit step if it belongs to a subject in the unknown unknown dataset. The system determines the prediction of \(u\) as follows:

1. Compute \(v_{*,u} = f(s_{*,u})\)
2. Find provisional subject \(p = \arg \min_a ||M_a - v_{*,u}||_2\)
3. If \(h_p(v_{*,u}) \geq \tau\), then “\(u\) is recognized as \(p\)”
4. Otherwise, “\(u\) is not recognized”

where \(\tau\) is one of the hyper-parameters of the system. A conceptual diagram of the test phase is illustrated in Figure 4.

![Figure 4](image-url)

**Figure 4.** Illustration of gait recognition using the trained model. In the example, unit step \(s_{*,u}\) is recognized as that of the “green” subject, whereas unit step \(s_{*,w}\) is not recognized.

3. Experiment

Using empirical datasets, we demonstrate the recognition accuracy of our proposed method with distinct sensing modalities (single and triple) and different network architectures (CNN, RNN, and ensemble).

3.1. Datasets and Evaluation Metric

We gathered gait information data from 30 adults aged 20 to 30 years. The insole was used to collect the data while the subjects walked for approximately 3 minutes. The data that were gathered during this time included approximately 151 unit steps on average per subject, and the entire dataset consisted of 4544 unit steps. In the experiment, we set the standard length to be \(d = 87\).

As shown in Figure 5, we split the data into three sets—training, unknown known, and unknown unknown. First, we randomly selected 16 out of the 30 subjects, and allocated 100% of the unit steps to the training dataset, which was used to train the CNN, RNN, and ensemble models independently. Second, we selected 7 out of the remaining 14 subjects arbitrarily. For each subject among the selected people, \(n = 10\) unit steps were utilized to train the OSVM algorithm and the decision boundary of
the subject was determined. Except for these \( n \) unit steps, all unit steps of the selected 7 subjects are allocated to the unknown known test dataset. Finally, all unit steps of the remaining 7 subjects are allocated to the unknown unknown test dataset \[ [17] \]. The number of unit steps in the training dataset is approximately 2423, and the number of unit steps of the unknown known test and the unknown unknown test datasets were approximately 990 and 1060, respectively. We repeated generating the datasets 20 times. For each dataset, we trained and tested the network independently and reported the averaged evaluation metrics. For a unit step in the unknown known test dataset, we define a true positive (TP) if a unit step is recognized correctly, and a false negative (FN) otherwise. In contrast, for a unit step in the unknown unknown test dataset, we define a true negative (TN) if a unit step is not recognized as any subject in the unknown known test dataset, and a false positive (FP) otherwise. We report the true positive rate \( TPR = \frac{TP}{TP+FN} \), the true negative rate \( TNR = \frac{TN}{TN+FP} \), and the accuracy \( ACC = \frac{TP+TN}{TP+FN+TN+FP} \).

![Figure 5. Illustration of the approach we used to split the data into training, unknown known test, and unknown unknown test datasets.](image)

3.2. Multi-Modal Sensing

The distributions of ACC as a function of \( \gamma \) and \( \nu \) for the CNN, RNN, and ensemble models are shown in Figure 6a. Clearly, selecting \( \gamma \) and \( \nu \) is critical to the overall recognition accuracy of the models. A comparison of the area in which the rates are greater than 90\% (light green to yellow areas) indicates that the region of the ensemble model is broader than that of the regions of the CNN or RNN model. This means that the ensemble model has a weak dependency when selecting \( \gamma \) and \( \nu \), which affects the robustness of the recognition result. The distribution of the TPR is shown in Figure 6b. A comparison of the area in which the rates are greater than 93\% (yellow), the region of the RNN model is slightly broader than that of the CNN model. The overall distribution of the ensemble model is similar to that of the RNN model. The distribution of the TNRs is shown in Figure 6c. Contrary to the distributions of the TPR, the overall distribution of the ensemble model is almost identical to the distribution of the CNN model. In particular, a comparison of the area in which the rates are greater than 93\% (yellow) reveals that the region of the CNN model is significantly broader than that of the RNN model. These distributions of the TNR explain why the ACC of the RNN model is significantly lower than the ACC of the CNN model. Utilization of the proposed system in a practical application would require the hyperparameters to be tuned by considering both the TPR and TNR at the same time. For example, if the system was to reject all unit steps, then we could achieve 100\% in TNR, but the TPR would equal 0\%. In this sense, we set the hyperparameters to minimize the differences between TPR and TNR.
Figure 6. Comparison of the ACC, TPR, and TNR results as a function of $\gamma$ and $\nu$ for the ensemble, convolutional neural network (CNN), and recurrent neural network (RNN). The same color denotes a similar rate (maximum 1% difference), with yellow indicating the highest rates.

To determine the effect of $\tau$, we specified separate values of $\gamma$ and $\nu$ for the different models in the following experiment. We used $\gamma = 1.9$ and $\nu = 0.06$ for the ensemble model, $\gamma = 1.8$ and $\nu = 0.06$ for the CNN model, and $\gamma = 2.2$ and $\nu = 0.08$ for the RNN model. In Figure 7, we see that choosing a $\tau$ value smaller than 0 significantly improves the TPR and ACC. Based thereupon, we propose alternative options for choosing $\tau$ instead of $\tau = 0.0$ for the decision boundary in the latent space.
Figure 7. Performance as function of $\tau$ for fixed $\gamma$ and $\nu$. We set $\gamma = 1.9$ and $\nu = 0.06$ for the ensemble model, $\gamma = 1.8$ and $\nu = 0.06$ for the CNN model, and $\gamma = 2.2$ and $\nu = 0.08$ for the RNN model.

3.3. Uni-Modal Sensing

To determine the contribution of each sensing modality to the accuracy, we trained and tested the models using uni-modal sensing. Effectively, in each sensing modality, only the corresponding sub-network was activated, whereas the two other sub-networks were deactivated while the network was being trained and tested. The TPR, TNR, and ACC of the uni-modal ensemble model as function of $\tau$ are compared in Figure 8. The overall performance of the ensemble model using pressure sensing was slightly lower than that of the others.

Figure 8. Performance of the uni-modal ensemble model as function of $\tau$ for fixed $\gamma = 1.9$ and $\nu = 0.06$.

Figure 9 compares the accuracy of unimodal sensing and multimodal sensing. In the case of the acceleration sensing modality, all the network models (Ensemble, CNN, RNN) showed the best performance compared to the other sensing modalities with the pressure sensing modality being the worst. In particular, the difference between these modalities is noticeable when the RNN model is used. Detailed TPR, TNR, and ACC results obtained with all the network models for multimodal and unimodal sensing are summarized in Table 1.
Table 1. Performance of the network models using multi-modal and uni-modal sensing. The blue color denotes the best accuracy of each network model.

| Model | Sensing | $\gamma$ | $\nu$ | $\tau$ | TPR   | TNR   | ACC   |
|-------|---------|----------|-------|--------|-------|-------|-------|
|       |         |          |       | -0.1   | 0.9342| 0.9375| **0.9360**|
| Ensemble | Multi   | 1.9      | 0.06  | -0.09  | 0.8889| 0.8845| 0.8876|
|        | Pressure|         |       | -0.08  | 0.8871| 0.9087| 0.8985|
|        | Acceleration |     |       | -0.1   | 0.8965| 0.8895| 0.8930|
|        | Rotation |         |       | -0.1   | 0.8965| 0.8895| 0.8930|
| CNN   | Multi   | 1.8      | 0.06  | -0.08  | 0.8803| 0.8802| 0.8808|
|       | Pressure|         |       | -0.09  | 0.8892| 0.8788| 0.8840|
|        | Acceleration |     |       | -0.08  | 0.8705| 0.8919| 0.8816|
|        | Rotation |         |       | -0.08  | 0.8705| 0.8919| 0.8816|
| RNN   | Multi   | 2.2      | 0.08  | -0.1   | 0.8745| 0.8759| **0.8757**|
|       | Pressure|         |       | -0.07  | 0.7752| 0.7760| 0.7757|
|        | Acceleration |     |       | -0.1   | 0.8173| 0.8283| 0.8224|
|        | Rotation |         |       | -0.1   | 0.8015| 0.8221| 0.8129|

The recognition accuracy in the previous papers [13,38] were ranging from 98.5% to 99.5%, which is higher than this study’s result. However, a direct comparison is inappropriate due to the different problem setting (for example, addressing the closed set problem [13]), or different datasets and devices (for example, using the unknown known test dataset only collected by smartphones [38]).

4. Discussion

To verify that the system forms a discriminative cluster for each subject, we present the t-SNE [47] plots of the embedding vectors of the unit steps in the unknown known and the unknown unknown test dataset in Figure 10. Considering that the networks were trained with subjects’ unit steps in the training set only, these plots show that the proposed system learns the general characteristics of unknown subjects’ gait patterns satisfactorily.
Figure 10. t-SNE plots of embedding vectors of subjects in the unknown known and the unknown unknown test datasets with multi-modal sensing. Each subject is represented by a unique color.

To enable us to quantitatively analyze our results, we devised a distance function between two unit steps using their embedding vectors. This distance function is defined by

\[ d(s, s') = ||f(s) - f(s')||_2 = ||v - v'||_2. \]  

(4)

The distributions of the distances between homogeneous and heterogeneous unit steps, respectively, are plotted in Figure 11. The blue line shows the distribution of the distances between homogeneous unit steps, which are two unit steps of identical subjects, and the orange line shows the distribution of distances between heterogeneous unit steps, which are two unit steps of different subjects. A clear distinction between the two distribution curves would signify the recognition accuracy of the system to be outstanding. Unfortunately, the two curves overlap to a certain extent, indicating that potential recognition errors may occur.

Figure 11. Distributions of distances between homogeneous unit steps and between heterogeneous unit steps in the latent space.

5. Conclusions

We proposed a new framework to recognize people based on their gait information. The proposed framework is the first approach to address the complete open set gait recognition from the data collected using wearable devices, namely insoles. Assuming an environment in which all people would be wearing the insole, our proposed framework could be applicable to variety of functions, for example, user verification. To build a user verification system, the system administrator would need to collect gait information for only 10 cycles of walking for every user. This would enable the system to recognize a user by examining a single cycle of their walking with 93.6% accuracy. Because the system does not require the encoder networks to be trained every time users are added, our proposed framework is highly scalable. In the future study, we aim to improve the recognition accuracy by
minimizing overlap between the distributions of the distances of homogeneous and heterogeneous unit steps.

**Author Contributions:** Conceptualization, methodology, funding acquisition, and writing—review and editing: J.M., and S.-I.C.; Software, formal analysis, investigation, data curation and validation, and visualization: N.A.L. and N.H.M.; Writing—original draft preparation: J.M.; Project administration: S.-I.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** The present research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2018R1A2B6001400) and the Republic of Korea’s 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).

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

**References**

1. Johansson, G. Visual perception of biological motion and a model for its analysis. *Percept. Psychophys.* **1973**, 14, 201–211. [CrossRef]
2. Cutting, J.E.; Kozlowski, L.T. Recognizing friends by their walk: Gait perception without familiarity cues. *Bull. Psychon. Soc.* **1977**, 9, 353–356. [CrossRef]
3. Cutting, J.E.; Proffitt, D.R.; Kozlowski, L.T. A biomechanical invariant for gait perception. *J. Exp. Psychol. Hum. Percept. Perform.* **1978**, 4, 357. [CrossRef]
4. Manap, H.H.; Tahir, N.M.; Yassin, A.I.M. Statistical analysis of parkinson disease gait classification using Artificial Neural Network. In Proceedings of the 2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao, Spain, 14–17 December 2011; pp. 60–65.
5. Wahid, F.; Begg, R.K.; Hass, C.J.; Halgamuge, S.; Ackland, D.C. Classification of Parkinson’s disease gait using spatial-temporal gait features. *Inst. Electr. Electron. Eng. J. Biomed. Health Inform.* **2015**, 19, 1794–1802. [CrossRef]
6. Zeng, W.; Wang, C. Classification of neurodegenerative diseases using gait dynamics via deterministic learning. *Inf. Sci.* **2015**, 317, 246–258. [CrossRef]
7. Gao, J.; Cui, Y.; Ji, X.; Wang, X.; Hu, G.; Liu, F. A Parametric Identification Method of Human Gait Differences and its Application in Rehabilitation. *Appl. Sci.* **2019**, 9, 4581. [CrossRef]
8. Dehzangi, O.; Taherisadr, M.; ChangalVala, R. IMU-based gait recognition using convolutional neural networks and multi-sensor fusion. *Sensors* **2017**, 17, 2735. [CrossRef]
9. Connor, P.; Ross, A. Biometric recognition by gait: A survey of modalities and features. *Comput. Vis. Image Underst.* **2018**, 167, 1–27. [CrossRef]
10. Choudhury, S.D.; Tjahjadi, T. Silhouette-based gait recognition using Procrustes shape analysis and elliptic Fourier descriptors. *Pattern Recognit.* **2012**, 45, 3414–3426. [CrossRef]
11. Cheng, M.H.; Ho, M.F.; Huang, C.L. Gait analysis for human identification through manifold learning and HMM. *Pattern Recognit.* **2008**, 41, 2541–2553. [CrossRef]
12. 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. [CrossRef]
13. Moon, J.; Minaya, N.H.; Le, N.A.; Park, H.C.; Choi, S.I. Can Ensemble Deep Learning Identify People by Their Gait Using Data Collected from Multi-Modal Sensors in Their Insole? *Sensors* **2020**, 20, 4001. [CrossRef]
14. Muaaz, M.; Mayrhofer, R. Smartphone-based gait recognition: From authentication to imitation. *IEEE Trans. Mob. Comput.* **2017**, 16, 3209–3221. [CrossRef]
15. Wan, C.; Wang, L.; Phoha, V.V. A survey on gait recognition. *ACM Comput. Surv.* (CSUR) **2018**, 51, 1–35. [CrossRef]
16. Scheirer, W.J.; de Rezende Rocha, A.; Sapkota, A.; Boulou, T.E. Toward open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, 35, 1757–1772. [CrossRef] [PubMed]
17. Geng, C.; Huang, S.J.; Chen, S. Recent advances in open set recognition: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2020.
18. Choi, S.I.; Lee, S.S.; Park, H.C.; Kim, H. Gait type classification using smart insole sensors. In Proceedings of the TENCON 2018—2018 IEEE Region 10 Conference, Jeju, Korea, 28–31 October 2018; pp. 1903–1906.
19. Murray, M.P.; Drought, A.B.; Kory, R.C. Walking patterns of normal men. J. Bone Jt. Surg. 1964, 46, 335–360. [CrossRef]
20. Lee, S.S.; Choi, S.T.; Choi, S.I. Classification of Gait Type Based on Deep Learning Using Various Sensors with Smart Insole. Sensors 2019, 19, 1757. [CrossRef]
21. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8 June 2015; pp. 815–823.
22. Schölkopf, B.; Williamson, R.C.; Smola, A.J.; Shawe-Taylor, J.; Platt, J.C. Support vector method for novelty detection. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 1 January 2000; pp. 582–588.
23. Niyogi, S.A.; Adelson, E.H. Analyzing and recognizing walking figures in XYT. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; Volume 94, pp. 469–474.
24. Świtoński, A.; Polański, A.; Wojciechowski, K. Human identification based on gait paths. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Ghent, Belgium, 22–25 August 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 531–542.
25. Yu, T.; Zou, J.H. Automatic human Gait imitation and recognition in 3D from monocular video with an uncalibrated camera. Math. Probl. Eng. 2012, 22–26.
26. Zhang, Z.; Tran, L.; Yin, X.; Atoum, Y.; Liu, X.; Wan, J.; Wang, N. Gait Recognition via Disentangled Representation Learning. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16 June 2019; pp. 4710–4719.
27. Yogarajah, P.; Chaurasia, P.; Condell, J.; Prasad, G. Enhancing gait based person identification using joint sparsity model and L1-norm minimization. Inf. Sci. 2015, 308, 3–22. [CrossRef]
28. 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. Sci. 2017, 7, 210. [CrossRef]
29. Lenac, K.; Sušanj, D.; Ramakić, A.; Piničić, D. Extending Appearance Based Gait Recognition with Depth Data. Appl. Sci. 2019, 9, 5529. [CrossRef]
30. Chen, X.; Xu, J.; Weng, J. Multi-gait recognition using hypergraph partition. Mach. Vis. Appl. 2017, 28, 117–127. [CrossRef]
31. Chen, X.; Weng, J.; Lu, W.; Xu, J. Multi-gait recognition based on attribute discovery. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 40, 1697–1710. [CrossRef] [PubMed]
32. Bodor, R.; Drenner, A.; Fehr, D.; Masoud, O.; Papanikolopoulos, N. View-independent human motion classification using image-based reconstruction. Image Vis. Comput. 2009, 27, 1194–1206. [CrossRef]
33. Hu, M.; Wang, Y.; Zhang, Z.; Little, J.J.; Huang, D. View-invariant discriminative projection for multi-view gait-based human identification. IEEE Trans. Inf. Forensics Secur. 2013, 8, 2034–2045. [CrossRef]
34. 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, 209–226. [CrossRef] [PubMed]
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3 December 2012; pp. 1097–1105.
36. Liu, L.; Peng, Y.; Wang, S.; Liu, M.; Huang, Z. Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors. Inf. Sci. 2016, 340, 41–57. [CrossRef]
37. el Achkar, C.M.; 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. [CrossRef] [PubMed]
38. Gadaleta, M.; Rossi, M. Idnet: Smartphone-based gait recognition with convolutional neural networks. Pattern Recognit. 2018, 74, 25–37. [CrossRef]
39. Cortes, C.; Vapnik, V. Support-vector networks. Mach. Learn. 1995, 20, 273–297. [CrossRef]
40. Choi, S.I.; Moon, J.; Park, H.C.; Choi, S.T. User Identification from Gait Analysis Using Multi-Modal Sensors in Smart Insole. Sensors 2019, 19, 3785. [CrossRef] [PubMed]
41. Cevikalp, H.; Neamtu, M.; Wilkes, M.; Barkana, A. Discriminative common vectors for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 4–13. [CrossRef] [PubMed]
42. Alotaibi, M.; Mahmood, A. Improved gait recognition based on specialized deep convolutional neural network. *Comput. Vis. Image Underst.* **2017**, *164*, 103–110. [CrossRef]

43. Footlogger Insole. Available online: [http://footlogger.com/hp_new/?page_id=11](http://footlogger.com/hp_new/?page_id=11) (accessed on 20 October 2020).

44. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11 April 2011; pp. 315–323.

45. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef][PubMed]

46. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

47. Maaten, L.v.d.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

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

[cc-by-4.0](http://creativecommons.org/licenses/by/4.0/) © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).