

Article

Predicting Human Motion Signals Using Modern Deep Learning Techniques and Smartphone Sensors

Taehwan Kim, Jeongho Park, Juwon Lee and Jooyoung Park *

Department of Control and Instrumentation Engineering, Korea University, 2511 Sejong-ro, Sejong-City 30019, Korea; kteaw0110@korea.ac.kr (T.K.); seanpark0107@korea.ac.kr (J.P.); saero94j@korea.ac.kr (J.L.)

* Correspondence: parkj@korea.ac.kr; Tel.: +82-10-9003-1810

Abstract: The global adoption of smartphone technology affords many conveniences, and not surprisingly, healthcare applications using wearable sensors like smartphones have received much attention. Among the various potential applications and research related to healthcare, recent studies have been conducted on recognizing human activities and characterizing human motions, often with wearable sensors, and with sensor signals that generally operate in the form of time series. In most studies, these sensor signals are used after pre-processing, e.g., by converting them into an image format rather than directly using the sensor signals themselves. Several methods have been used for converting time series data to image formats, such as spectrograms, raw plots, and recurrence plots. In this paper, we deal with the health care task of predicting human motion signals obtained from sensors attached to persons. We convert the motion signals into image formats with the recurrence plot method, and use it as an input into a deep learning model. For predicting subsequent motion signals, we utilize a recently introduced deep learning model combining neural networks and the Fourier transform, the Fourier neural operator. The model can be viewed as a Fourier-transform-based extension of a convolution neural network, and in these experiments, we compare the results of the model to the convolution neural network (CNN) model. The results of the proposed method in this paper show better performance than the results of the CNN model and, furthermore, we confirm that it can be utilized for detecting potential accidental falls more quickly via predicted motion signals.



Citation: Kim, T.; Park, J.; Lee, L.; Park, J. Predicting Human Motion Signals Using Modern Deep Learning Techniques and Smartphone Sensors. *Sensors* **2021**, *21*, 8270. <https://doi.org/10.3390/s21248270>

Academic Editors: David Silvera-Tawil, Qing Zhang and Mahnoosh Kholghi

Received: 18 November 2021

Accepted: 8 December 2021

Published: 10 December 2021

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



Copyright: © 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: wearable sensors; human motion; prediction; recurrence plot; Fourier neural operator

1. Introduction

The world-wide use of smartphones offers not only the convenience of accessing and searching information anywhere, but also various applications to enhance the quality of life. As smartphones have developed, various sensors have been included in smartphones, and a variety of accompanying applications have been introduced. Recently, in addition to basic IMU sensors, various sensors such as photoresistors, fingerprint sensors, and GPS have been added to smartphones, and the performance of the cameras has also improved, so that various studies using these can be conducted. In this paper, we intend to deal with healthcare-related problems utilizing these smartphone sensors. Various research topics of interest in healthcare include analyzing human behavior, and recent studies have been conducted on recognizing and characterizing human motion patterns [1–11]. Various previous studies have examined the recognition or analysis of human motion, and a number of studies have examined recognition of human activities via machine learning [5–9]. Of particular interest is research on aspects of human gait, e.g., while walking and running [12]. Many studies have been conducted on motion analysis using classical to modern machine learning methodologies, such as Gaussian processes [2,3], to the latest deep learning techniques [4–7,9–11]. Wang, J. et al. [2] proposed Gaussian process dynamical models to characterize human motion from captured motion data. Their model was constructed with

latent space to represent the dynamics of motion and observation space, and the model showed remarkable results in the latent space. Kim, T. et al. [3], and Kim, J. et al [4] used an accelerometer and a gyro sensor to characterize human movement in latent space. In one study [3] the proposed model could be seen as a combination of the variational autoencoder and a Gaussian process, and the detection of risks such as falling was examined by using the characterized latent space. One particular study [5] showed suitable performance in human activity recognition tasks using magnetic induction and recurrent neural networks, and in another study [7], human activity recognition was achieved by converting signals obtained from triaxial accelerometers into images. These studies sometimes used camera sensors, or sometimes wearable sensors to obtain health signals when attached directly to individuals to analyze and predict movements. Typically, inertial measurement units (IMUs), such as accelerometers and gyroscopes, have been widely used to measure health signals. These sensor signals are generally in the form of time series, and many methods exist for dealing with this data. A spectrogram, which is represented by combining properties of waveforms and spectra, is a widely used method to convert time series data to an image type, as it visualizes changes in amplitude with time and frequency. Numerous previous studies have used spectrograms for recognizing human activities [13–16]. In addition to spectrograms, many other techniques exist to render time-series data as image formats. The easiest method to transform sequential data into image data is to image the signal itself, and another technique is to render the three axes of signals obtained from sensors directly into a corresponding RGB format [17]. Zheng, X. et al [17] compared the performance of various signal processing methods such as raw plots and multichannels for human activity recognition. A method of imaging real-valued signals from sensors by dividing them into integers and decimal places has also been introduced [18]. A recurrence plot is another popular method for processing time-series data as images, and some studies have been conducted on recognizing human activities with such recurrence plots [19–22]. Specifically, recurrence plots can show the global and local contextual features of time-series data, and through these features, temporal information about the data can be captured.

Recently, notable results have been demonstrated in many fields using deep learning methods. In the field of computer vision, deep learning models that perform better than humans in simple classification problems have been researched. Such methods have also demonstrated excellent performance in studies of topics such as object detection and segmentation tasks, and in even in natural language processing, deep learning models have been used to show results that are almost similar to or superior to human performance. In such studies, the use of convolutional neural networks (CNNs) and fully connected networks have generally shown solid performance in video and image fields, and recurrent networks such as RNN, LSTM [23], and GRU [24] have shown notable results in speech and natural language processing. For example, recent studies have reported suitable performance in speech and natural language processing with CNN [25], and also noteworthy outcomes in video and image fields with the transformer model, which is a similar series of recurrent networks designed for 1D sequence data processing [26,27]. Hence, the use of deep learning models has traversed the boundaries of domains like video and audio, making for a wide range of applications beyond those of specific networks that have shown excellent performance in specific fields. In addition to this expansion of applications of deep learning models, other studies in a different vein have been undertaken. In addition to major research fields using deep learning to determine decision boundaries or to estimate functions through regression, deep learning has recently been used to solve differential equations [28–31]. Euler's methods for solving ordinary differential equations(ODEs) are similar to those used by recurrent neural networks such as RNN, LSTM, and GRU [31]. For such reasons, as in [31], studies have been conducted on solving ordinary differential equation with neural networks, and, in [28–30], neural networks were used to solve partial differential equations (PDEs). Specifically, the study [30] proposed a method, Fourier neural operator (FNO), to solve the PDE problem by combining neural network architecture and the Fourier transform. The architecture introduced in the study [30] can be understood as a

type of CNN, but this method also has the properties of mesh-independence, in contrast to traditional, fundamental CNN. This feature means that when given an input of type $N \times N$ rather than the size of the learned input, it works even if the size of the input is an FNO model trained with $M \times M$. Therefore, as long as the model is trained with one resolution, inferences can be made even with other resolutions without retraining. Based on these advantages and features, this method is utilized in this study to predict human motion signals, and we show that this leads to performance superior to that of the CNN architecture through our experimental results.

In this paper, research on human motion analysis was conducted, with the primary objective of presenting a method for predicting human motion signals using modern deep learning techniques and smartphone sensors. Our method relies on the FNO model for its excellent representational capability in the frequency domain, and recurrence plots for their efficiency in handling local and global features. Recurrence plots were originally proposed as a visualization tool for analyzing characteristics of dynamical systems. Recently, many studies have applied these methods to the task of recognizing human activities by making use of the characteristics of recurrence plots, as they are able to capture temporal patterns over small scale and large scale patterns. However, through previous studies [19,20,32,33], it was found that much information loss may occur due to the threshold and step functions in the original recurrence plot formulation. To avoid this problem, some studies used only the distance component to construct recurrence plots [20,32]. Referring to these previous studies, we also use the distance component of the original recurrence plot to convert sensor data into 2D image data, and apply this to the modern deep learning methodology, FNO, to predict a user's future movements. For more details of the recurrence plot, and how it is used in our experiments to handle sensor signals as a visualization, one may refer to Section 2.1 below. In the recurrence plot, not only small scale patterns but also large scale patterns are revealed, so in order to fully utilize the recurrence plot, we consider use of the FNO model, which can capture the global context using the Fourier space. The overview of our method for estimating sensor signals is sketched out in Figure 1. Using the method presented in this paper, we also explored the possibility of using it for risk detection, such as for falls.

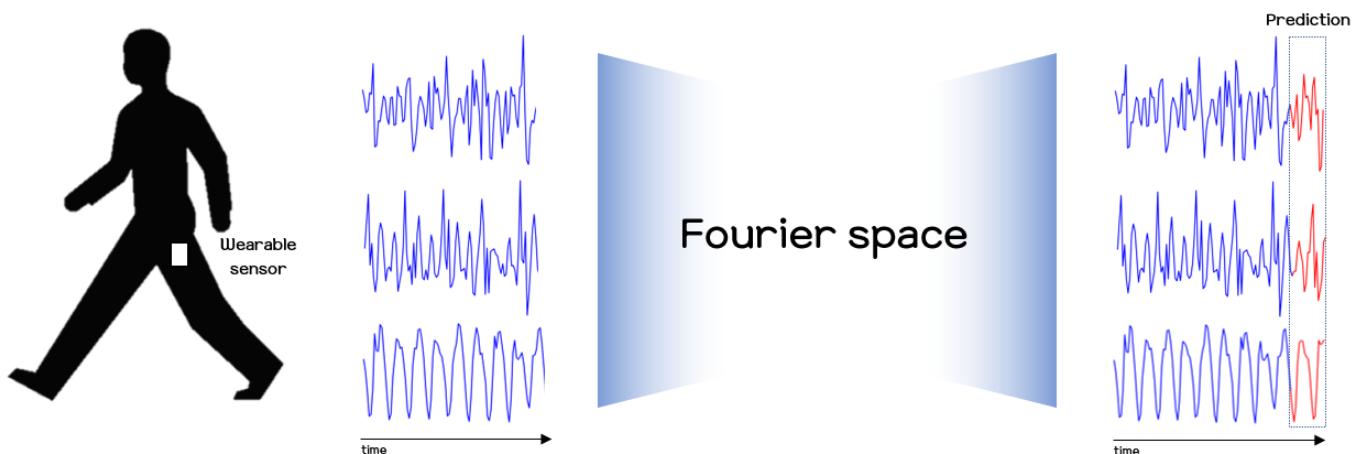


Figure 1. Overview of our proposed method for predicting human motion using wearable sensors and the modern deep learning model.

The paper is structured as follows: In Section 1, we briefly explain the rationale for this research and the methods utilized, and introduce related works. In Section 2, as a method for predicting wearable sensor signals, we will introduce the latest deep learning model and the conversion of time series data into a form suitable for application to the model. In the next portion, Section 3, we present the process for acquiring the data used in the experiments, and show the results of the experiments with comparisons. The final

Section 4 includes conclusions and discussion of applications for the methods presented in this paper.

2. Methods

In this section, we introduce the methods used in this paper. As mentioned in the previous section, we develop an estimation of a user's activity signals by means of wearable sensors that are built into the user's smartphone. Among the many kinds of sensors in a smartphone, we utilize the IMU sensor to forecast human motion signals. The acquired raw sensor signals are time-series forms of data, then we transfer the signals to an image format. After converting the data, we use the recently introduced deep neural network known as a Fourier neural operator [30], hereafter referred to as the FNO model, to predict human motion. We consider the recurrence plots [34], which is powerful for converting time-series data to image data. Recurrence plots generally deal with all time steps at once, but we split the time steps with a time window and then use this as the short-term recurrence plot. Figure 2 shows the overall outline of our method used in this paper.

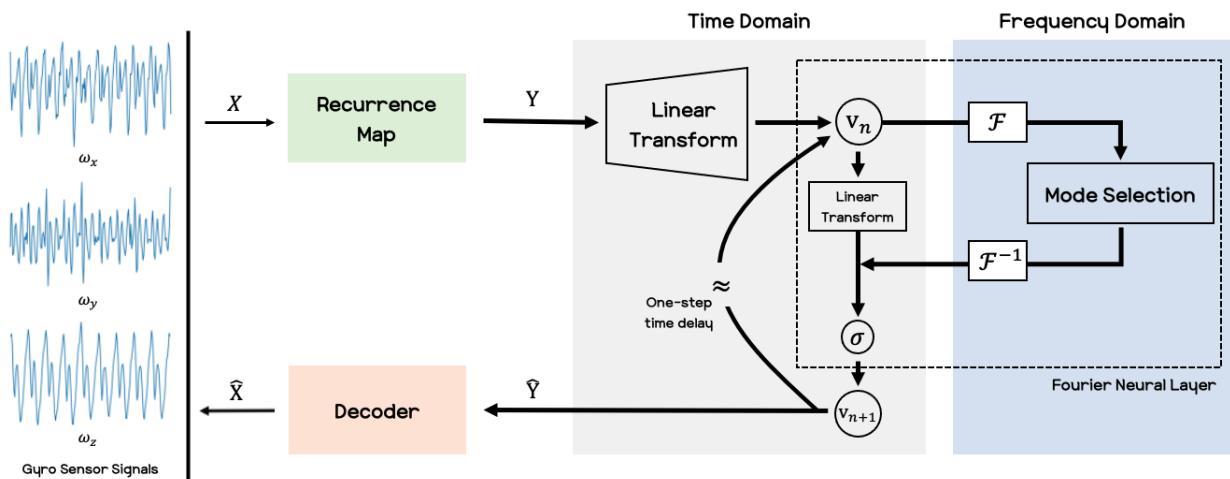


Figure 2. Pipeline schematic of our proposed method for predicting upcoming human motion signals. The linear transform lifts the recurrence plot to a high-dimensional representation or performs a linear combination of the features. A Fourier neural layer composed of a Fourier transform, \mathcal{F} , and its inverse, \mathcal{F}^{-1} , crosses between the time domain and the frequency domain, and it extracts features by removing unnecessary frequencies from the mode selection. Then the output of the Fourier neural layer is given as v_{n+1} which is applied to the sigmoid function, σ .

2.1. Recurrence Plot

Various studies have recognized and attempted to predict human activities, using not only raw sensor signals themselves via normalization, but also raw sensor signals as transformed images, such as raw plots, multi-channels [17], spectrograms [13–16], and recurrence plots [19–22,34]. The raw plots method transforms raw sensor signal data directly to an image, i.e., plotted images along axes corresponding to the time and amplitude of the sensor signals. Another transformation, the multichannel method, renders the sensor signals with three axes, i.e., x-, y-, and z-axes, and translates them into the three color channel components (red, green, blue), in an RGB color format [17]. The spectrogram representation is widely used for dealing with time-series data from speech waveforms in acoustic modeling to IMU sensor data for human motion recognition. It provides a representation of changes in the energy content of a signal as a function of frequency and time [15] and thus shows advantages in classification accuracy and computational complexity [14]. In addition, many other methods exist, but in this paper, the recurrence plot is considered for preprocessing raw sensor signals.

Recurrence Plot The recurrence plot (RP) was originally a visualization tool to study complex dynamic systems, which was first proposed in [34]. A recurrence plot expresses information by using the difference between time i and time j in time series data. It can

thus represent time correlation, and based on differences between values at each time step, it is formed with small and large-scale patterns. The small and large patterns provided in a recurrence plot are represented as *texture* and *typology* respectively [34]. These patterns provide information about dynamical systems, and the information is displayed as diagonal lines, dots, vertical lines, and such in the recurrence plots. The recurrence matrix for the recurrence plot is computed as follows when time series data $X = [x_1, \dots, x_T]$ is given.

$$\text{Recurrence}(X) = \Theta(\epsilon - \|x_i - x_j\|) \quad \text{where } i, j = 1, \dots, T \text{ and } x_i \in \mathbb{R}^D \quad (1)$$

Here, ϵ and $\Theta(\cdot)$ are, respectively, a threshold value and the unit step function, which yields 0 or 1, depending on the given value. The component $\|\cdot\|$ is a norm, and a proper norm should be selected according to the task. The most commonly used norms are the L_1 norm, L_2 norm, and L_∞ norm. The general recurrence matrix has a square matrix form, as it represents only one dynamical system. Two typical variants of the recurrence plot can express multivariates, the cross recurrence plot and the joint recurrence plot.

Cross Recurrence Plot The cross recurrence plot (CRP) is one representative variant of the recurrence plot, and it is computed by comparing all states occurring in one dynamical system with all states occurring in another dynamical system. In the CRP, the two systems must have the same state dimension, but the length of time does not have to be the same, resulting in a cross recurrence matrix that may not be square. When two systems, $X = [x_1, \dots, x_{T_x}]$ and $Y = [y_1, \dots, y_{T_y}]$, are given and they are the same dimension, the cross recurrence matrix can be computed as follows.

$$\text{CrossRecurrence}(X, Y) = \Theta(\epsilon - \|x_i - y_j\|) \quad \text{where } i = 1, \dots, T_x, j = 1, \dots, T_y \text{ and } x_i, y_j \in \mathbb{R}^D \quad (2)$$

In order to generate proper inputs for machine learning models, in this paper, we follow the basic form of the recurrence plot, Equation (1), and use the L_2 norm. However, we did not use the threshold, ϵ , or the step function, Θ . In previous studies [19,20,32,33], it has been reported that when using the recurrence plot as the input of the deep neural network model, the thresholds and step functions can be sparse recurrence plots, resulting in information loss. In order to avoid such problems, we follow the strategy [20,32] of calculating recurrence plots without the step function and threshold. That is, we obtained recurrence plots using only the distance component of the recurrence plot.

$$\text{Recurrence}(X) = \|x_i - x_j\|_2 \quad \text{where } i, j = 1, \dots, T_w \quad (3)$$

In the experiments, we use smartphone sensors which are composed of three axes, and by using them, we conduct for predicting human motion signals. Our approach is similar to CRP, which builds its recurrence plots using multiple systems.

2.2. Fourier Neural Operator

Fourier Transform The Fourier transform is commonly used in image processing to decompose into components corresponding to sine and cosine. Filtering can be applied to an image by using the Fourier transform. For example, when a low-pass filter is applied to an image, the filtered image would appear blurred, with indistinct boundaries. On the other hand, a high-pass filter produces a sharp image with only the edges remaining. The low-frequency component is the portion of small differences in color between pixels, while color differences figure significantly in the high-frequency component. Many studies using Fourier transforms have been conducted on the universal approximation theorem [35], modern deep learning architecture [36], and other problems. Through these previous studies on applying Fourier transforms to neural networks, deep learning has been able to see advances in performance. Among these studies, one recent study directly combined a Fourier space with a neural operator architecture [30]. In this paper, we utilize this architecture to predict human motion signals.

Fourier Neural Operator Within the research on deep neural networks, a new type of neural architecture has been recently introduced, known as neural operators [28–30,37].

The architecture proposed in [30] is shown in Figure 3. The Fourier module in this figure performs convolution operations in the frequency domain through the Fourier transform, and then the inverse Fourier transform is applied to revert it to the spatial domain. The Fourier module in the Fourier layer are expressed as Equation (4).

$$F_n(x) = \mathcal{F}^{-1}(R_{\phi_n} \cdot (\mathcal{F}(x))) \quad \text{where } \forall x \in D \quad (4)$$

Here, \mathcal{F} and \mathcal{F}^{-1} are the Fourier transform and its inverse transform, respectively. R_{ϕ} is the parameter that functions like a kernel in convolution neural networks.

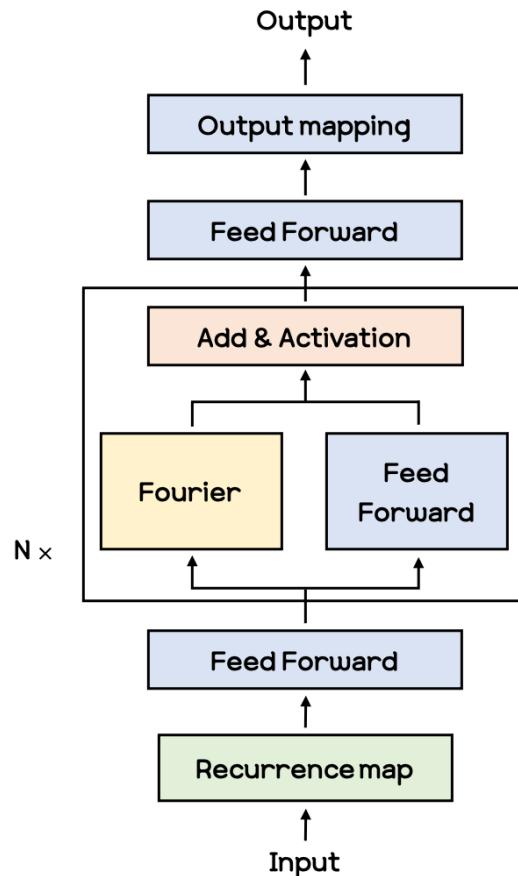


Figure 3. FNO architecture with N Fourier layers.

The Fourier module, $F_n(x)$, receives the high dimensional representation of inputs first and then receives the previous output of the Fourier layer for $N-1$ times. The Fourier layer consists of the Fourier module and a linear transformation, which can be expressed as Equation (5).

$$v_{n+1}(x) := \sigma(F_n(v_n(x)) + W_{\theta_n} v_n(x)), \text{ where } \forall x \in D \quad (5)$$

Here, σ is a non-linear activation function, and in our experiments we used the sigmoid function. We use this Fourier neural operator as a substitute for convolution neural networks. The entire process for training the Fourier neural operator using the recurrence plot data described in Section 2.1 is shown in Procedure 1.

Procedure 1 : Procedure for Performing Fourier Neural Operator

```

1: Initialize weights  $\theta$  and  $\phi$ .
2: Set parameters of prediction time step and the input time step,  $\hat{T}$  and  $T$ , respectively.
3: Get batch RP data for input,  $Y$ , and target,  $Y^{target}$ , through Procedure 2.

4: while converge do
5:    $v_0 = \text{Linear}_{\theta}(y_{t-T:t})$ 
6:   for  $n = 0, \dots, N-1$  do                                 $\triangleright$  Fourier neural layers
7:      $f_n = \mathcal{F}(v_n)$                                           $\triangleright$  frequency domain
8:     Select  $k$  frequency modes,  $f_n^k$ 
9:      $f_n^k = R_{\phi_n} \cdot f_n^k$ 
10:     $v_n^F = \mathcal{F}^{-1}(f_n^k)$ 
11:     $v_n^S = \text{Linear}_{\theta_n}(v_n)$                                 $\triangleright$  spatial domain
12:     $v_{n+1} = \sigma(v_n^F + v_n^S)$ 
13:   end for
14:    $\hat{y}_{t:t+\hat{T}} = \text{Linear}_{\theta}(v_N)$ 
15:    $\mathcal{L} = \text{MSE}(y_{t:t+\hat{T}}^{target}, \hat{y}_{t:t+\hat{T}})$ 
16:   Update weights,  $[\theta^{k+1}, \phi^{k+1}] \leftarrow [\theta^k, \phi^k] - \alpha \mathcal{L}$ 
17: end while

```

2.3. Decoder

In this paper, we use a decoder model to reconstruct the sensor signal from the recurrence plot data. Given the predicted outputs of the FNO model, the decoder model converts the recurrence plot images to the raw sensor signal. The architecture of the decoder used in the experiments is composed of two components, which are a long short-term memory (LSTM) and two fully connected layers. To train the decoder model, both the recurrence plot and raw sensor signals are used. Figure 4 shows the overall process of the proposed method in this paper. For training the FNO model and decoder model, three types of data need to be prepared: the input recurrence plot data as input to the FNO model, Y , and the target recurrence plot data that are to be predicted from the FNO model, Y^{target} . Lastly, the target sensor signals, X^{target} , are to be matched with the outputs predicted from the decoder model.

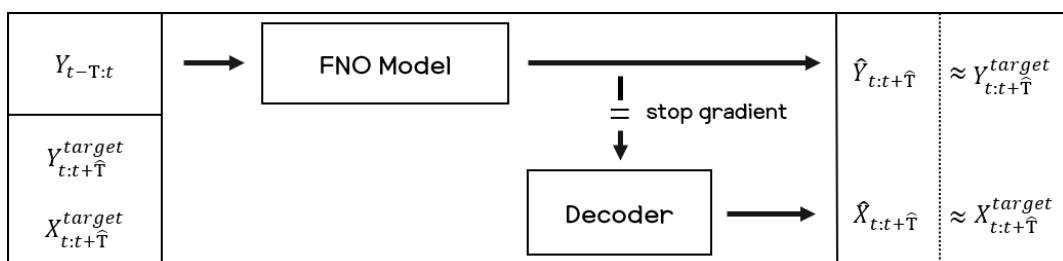


Figure 4. Block diagram for training the proposed model.

3. Experiments

3.1. Data Collection

For the experiments, we use a single smartphone and attached it to the user's right thigh. We refer to some previous studies [38–41] and public datasets, such as MobiAct [42] and UC-HAR [18], for the human activity recognition tasks. They used a smartphone and placed it in the trousers pocket to collect built-in sensor signals in the smartphone. We also considered placing the smartphone in the pocket based on the previous studies. When we put it in the trousers pocket and acquired sensor data, we found not only signals from the user's movement but also noise caused by moving around in the pocket in addition to the user's movements. For this reason, we fixed the smartphone near the right trousers pocket to obtain only the user's movement signal without any noise. The location and manner of fixation are shown in Figure 5.



Figure 5. Placement of the smartphone used for data collection.

The experiments were conducted at the Korea University R&D Center. To collect smartphone sensor signals, we used the Matlab application [43] and the iPhone XS model for detailed smartphone specifications, referring to the manufacturer's website [44]. A modern smartphone contains many sensors such as an accelerometer, gyroscope, and magnetometer. However, it has been known through previous studies that using the magnetometer should be used with caution due to problems such as direction dependence [45,46] and inefficiency in recognition performance [47,48]. Thus, in our experiments, we considered the use of accelerometers and gyroscopes, and we experimentally confirmed that using the gyroscope yields better performance than the accelerometer. Hence, we choose the gyroscope for acquiring the angular velocity corresponding to the user's motion. The angular velocity signal was acquired at 30 Hz, which was also experimentally selected between 10 Hz and 100 Hz. To obtain the gyro sensor signals at 30 Hz, we used the Matlab application, which can access the built-in sensors in the smartphone. In our experiments, Matlab was used only for data collection when the experiments were conducted. The data collection procedure consisted of the following: executing the Matlab application on the smartphone, entering the Sensors tab, and after setting the sensors to be used, pressing the start button to acquire sensor signals such as those from accelerometers, magnetometers, and gyroscopes, which are built into the smartphone. After pressing the start button on the Matlab application, the participants repeatedly performed pre-defined motions (e.g., running, walking), and after the motions ended, the signal data was automatically uploaded to a cloud server. We then downloaded the signal data and followed our procedure to predict the user's motions. For discriminating the walking and running motions in our experiments, we refer to the definitions [49] that one foot is always on the ground while walking, while both feet might simultaneously be off the ground during running. The whole process of obtaining sensor signal data from the smartphone is shown in Figure 6.

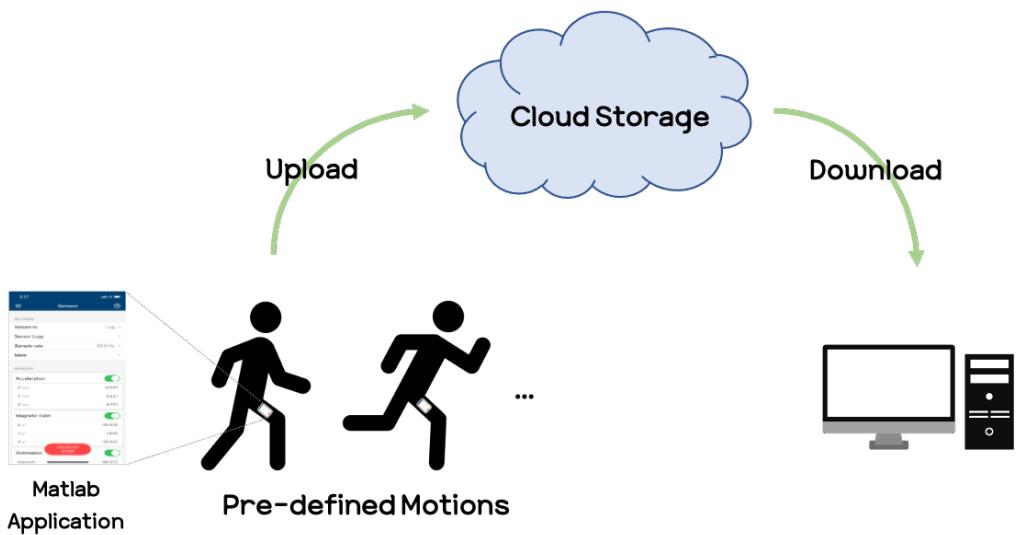


Figure 6. Process for acquiring sensor signals.

Also, the detailed procedure for obtaining the data is as follows:

- (a) Set the predetermined route for motions to be collected for the data (e.g., walking, running).
- (b) Place the smartphone on the thigh and tied it up to prevent shaking.
- (c) Execute the Matlab mobile application on the smartphone.
- (d) Set the sampling rate at 30 Hz, and set it to upload its sensor log to cloud storage.
- (e) Select the angular velocity sensor (among the acceleration, magnetic field, orientation, angular velocity, position sensor).
- (f) Press the start button to begin acquiring sensor data in the Matlab application.
- (g) The participants perform the predefined motion 3 s after pressing the start button to eliminate any effects that may have occurred before executing the action.
- (h) The participants perform the motion for about 60 s, which could total 1800 samples.
- (i) The participants ceases the motion and presses the stop button 3 s afterward, for the same reason of preventing noise related problems.
- (j) After identifying and naming the data set, the data are uploaded to the cloud server.
- (k) Download the data acquired from the gyro sensor to a desktop computer.
- (l) Repeat steps (d) to (k) for other motions.

The acquired signal consists of three axes of the gyro sensor, the x-, y-, and z-axes. Various methods for treating the x, y, and z axes for sensor data in recurrence plots have been introduced. In this paper, we utilized the strategy of concatenating the x, y, and z axes by referring to [22], where in the process of creating recurrence plots for the x, y, and z axes from a sensor, they first create recurrence plots for each axis and then concatenate the generated recurrence plots side by side. Although the order is slightly different, our approach follows essentially the same process in obtaining the recurrence plots. More precisely, we first create the X vector by concatenating the x, y, and z axes from a sensor and then create the recurrence plots for the X vector. We empirically found that the method of aligning the x, y, and z to construct the X vector and thereby creating the recurrence plot shows satisfactory performance. Let ω_x , ω_y , and ω_z be the raw sensor signals associated with each axis obtained from the smartphone sensor. We split the time sequence of each axis by using a time window for size, T_w , and stride s , and then combine the split windows on each axis as follows:

$$X_t = [\omega_{t-5}^x, \omega_{t-4}^x, \dots, \omega_t^x, \omega_{t-5}^y, \omega_{t-4}^y, \dots, \omega_t^y, \omega_{t-5}^z, \omega_{t-4}^z, \dots, \omega_t^z] \quad (6)$$

We converted the raw sensor signal to a recurrence plot with X_t . In the experiments, we set the window size and stride as 5 and 1, respectively. Thus, the sensor signals are

arranged in a dimensionality of 15 (=3 axis \times 5 time windows), and we additionally performed linear interpolation on the X_t with the Scipy.interpolation [50]. Note that when the frequencies of the input data are not similarly fixed, we need the capability of interpolation. For interpolation, we inserted one additional sample between the original samples, which are obtained from the sensor via linear interpolation until the last sample. After the interpolation, we transformed X_t into recurrence plot data according to Equation (3).

$$Y_t = \text{Recurrence}(X_t) \quad \text{where } X_t \in \mathbb{R}^{29} \quad (7)$$

The final resolution of the recurrence plot data is then 29×29 . In our experiments, we then attempted to predict the next few seconds from the FNO model by receiving the recurrence plot data for the past few seconds. We set the input time step, T , and the prediction time step, \hat{T} , as 30 and 20, respectively. That is, when the recurrence plot corresponding to 1 s is given as $Y_{t-30:t}$, the FNO model predicts samples, $Y_{t:t+20}$, associated with about 0.6 s of the recurrence plot. In the training phase, the 30 recurrence plots serve as an input to the FNO model, and the following 20 recurrence plots are targets that the model should predict. The whole process for producing the recurrence plot used in this paper is shown in Figure 7 and Procedure 2. Also, Figure 8 shows samples of the recurrence plot created by following the process of Procedure 2 for the motions considered in the experiment.

Procedure 2 : Procedure for Obtaining Recurrence Plot

- 1: Set parameters window size T_w .
 - 2: Acquire raw sensor signals from a wearable device (e.g., gyroscope): $\omega^x, \omega^y, \omega^z$.
 - 3: Set empty list to stack time window of each axis, D_x, D_y, D_z .
 - 4: **for** $t = T_w, \dots, \text{LENGTH}(\omega)$ **do** ▷ Stack the time windows of each axis
 - 5: $D^x \leftarrow D^x \cup \{\omega_{t-T_w}^x, \omega_{t-(T_w-1)}^x, \dots, \omega_t^x\}$
 - 6: $D^y \leftarrow D^y \cup \{\omega_{t-T_w}^y, \omega_{t-(T_w-1)}^y, \dots, \omega_t^y\}$
 - 7: $D^z \leftarrow D^z \cup \{\omega_{t-T_w}^z, \omega_{t-(T_w-1)}^z, \dots, \omega_t^z\}$
 - 8: **end for**
 - 9: Join split time window of each axis.
 - 10: **for** $t = 0, \dots, \text{LENGTH}(D)$ **do**
 - 11: $X \leftarrow X \cup \{[D_t^x; D_t^y; D_t^z]\}$
 - 12: **end for**
 - 13: Interpolate for each X_t .
 - 14: Calculate recurrence plot with X_t interpolated through Equation (3).
 - 15: **for** $t = 0, \dots, \text{LENGTH}(X)$ **do**
 - 16: $Y_t = \text{Recurrence}(X_t)$
 - 17: **end for**
-

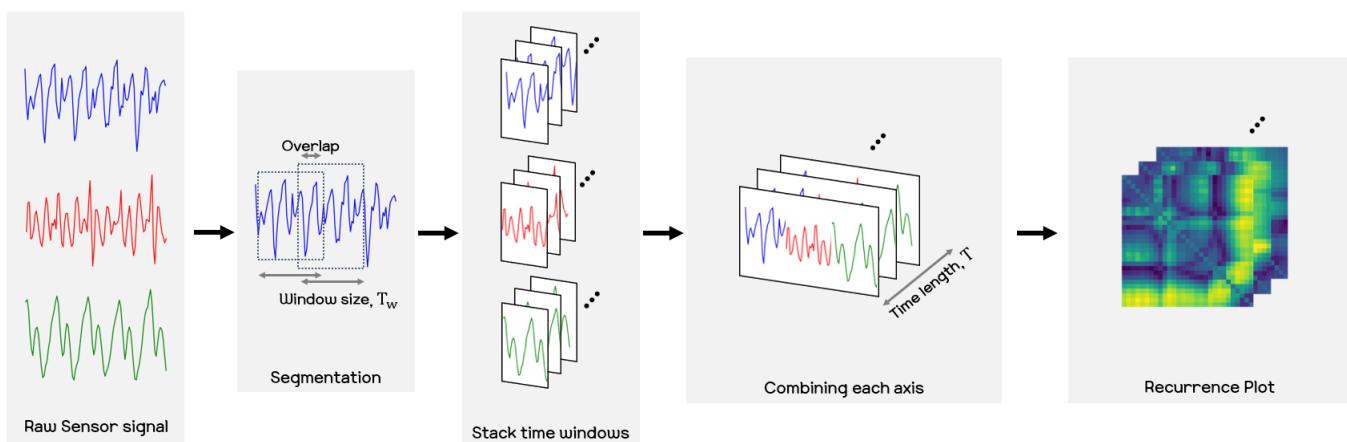


Figure 7. Procedure for converting raw sensor signals into a recurrence plot. Raw sensor signals are divided into time windows of size T_w and then converted into a recurrence plot by combining time windows corresponding to each x-, y-, and z-axis into one.

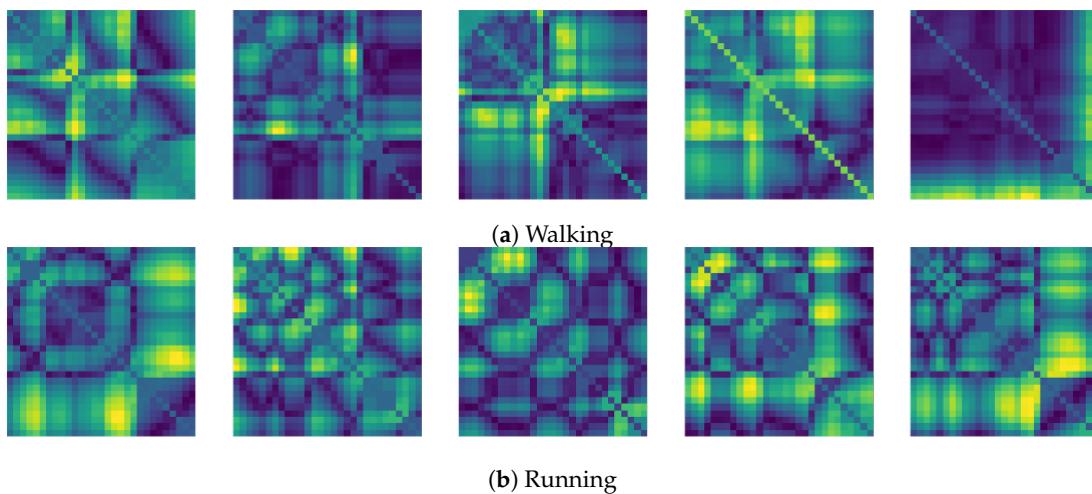


Figure 8. Recurrence plot samples for each motion from the gyro sensor.

In this paper, we introduce a method for predicting human motion signals using modern deep learning techniques and smartphone sensors. For this we deal with the two types of input data, for the FNO model and for the decoder network. The FNO model is trained with the recurrence plot data, which are converted from the raw sensor signals, and the decoder model is trained with both the recurrence plot and raw sensor signals to recover the raw sensor signals from the recurrence plot. Given $Y_{t-30:t}$, the input data consisting of the recurrence plot, the FNO model provides the predicted outputs, $Y_{t:t+20}$, which are also the recurrence plot type. We recover the recurrence plot data for the sensor signal type by means of the decoder network. The output of the decoder model would be then the signals, $X_{t:t+20}$, which follow the input signals corresponding to the input of the FNO model.

3.2. Experimental Results

This section reports on the experimental results of the proposed method for predicting human motion signals. After recurrence plot data are converted from the raw gyro sensor signals via the process described in Section 3.1, we use the FNO model to predict the upcoming motion signals. To verify the results of the FNO model from the recurrence plot data, we collected motion data from six volunteers, and each volunteer performed the experiment with walking and running motions. The sensor signals were collected for about 60 s at 30 Hz and converted into the recurrence plot data, and then the converted data are

entered as input into the FNO model, as described in Section 2.2. Through this process, we can obtain an estimated event sequence for a motion type based on the recurrence plot and FNO model, and then the data are converted into sensor signal form by the decoder model. In the training phase, we set the batch size and epoch at 32 and 200, respectively, and the number of Fourier neural layers in the FNO model was set at 4. The FNO model for recurrence plot prediction and the decoder for converting images into sensor signals are trained separately by means of the Adam optimizer [51]. We also use a learning rate schedule, such that the learning rate starts at 0.001 and decreases by half every 50 epochs. The mean squared error (MSE) loss is used as the loss to train both models. For training and analyzing the experimental results, we split the entire data set acquired from each volunteer into training, validation, and test sets, with a ratio of 6:2:2, respectively. In the data collection step, each motion was performed for about 1 min per subject, and the samples acquired by the gyroscope at this time are 1800 (30 Hz × 60 s). In addition, the data were collected twice for each motion per subject, and therefore, the total number of samples acquired is about 3600 per subject. We split these data in a 6:2:2 ratio, and the number of training, validation, and test recurrence plots then amount to about 2160, 720, and 720, respectively, per subject. When training the model, only the training set was used, and the validation set was used to check whether the model was overfitted. Figure 9 shows the MSE values of the FNO model for the test dataset that was conducted with walking motion. As noted above, we conducted the experiments with six subjects and trained the FNO model for each subject, so six models were trained. As seen in Figure 9, we can confirm that a personalized model was created for each subject. When tested with each individual test set, which was not used in the training phase, the MSE value is small, but when tested with other participants' data, the error value is relatively high. Thus, it can be seen that models trained on each individual subject can identify the subject. Also, Figure 10 shows the MSE values for running motion. The results for running motion also show that the models properly characterized each subject.

To compare the results of the FNO model, we trained another model, which is composed of convolutional neural layers, referred to as the CNN model for convenience in the rest of this paper. The CNN model consists of four convolutional neural layers and three fully connected layers, constructed in a manner similar to the FNO model. The kernel size and stride of each convolution layer in the CNN model are 5 and 1, respectively, and zero padding was added so that the image size would not change while passing through the convolutional neural layers. Each convolution layer was followed by a ReLU activation function, except for the final layer. That is, according to the FNO structure, the experiment was conducted by replacing the Fourier neural layer with the convolutional neural layer, while all other layers were the same. With the same learning parameters, optimizer, and learning rate scheduler as in the FNO model, the CNN model was trained. The results of the CNN model in Figure 11 indicate that the CNN model had higher MSE values than the FNO model for the same dataset. For example, the first diagonal entries of Figures 9 and 11a show that the MSE values of the FNO model and CNN model for subject #1 are 0.1577 and 0.2445, respectively. Furthermore, the first diagonal entries of Figures 10 and 11b show that the MSE values of the FNO model and CNN model for subject #1 are 0.1421 and 0.3141, respectively. As shown in Figures 9–11, the MSE values of the CNN model turn out to be higher than those of the FNO model throughout all subjects. From these results, we can confirm that the FNO model can better represent human motions expressed by converting sensor signals into an image format than a simple CNN model. Figures 12a and 13a show some samples of the predicted results of the FNO model for walking and running motions, respectively. These figures show that the predicted results almost equal the target recurrence plot images, in Figures 12b and 13b. On the other hand, in Figures 12c and 13c, in comparing the results with the FNO model, some of the CNN model samples are also shown, and those CNN results show lower predictive performance. With these results, the validity of the method for predicting human motion using the FNO model can be empirically confirmed. We additionally utilized the learning curves

for the training and validation dataset during the training process to check whether the model overfits. Figures 14 and 15 show some of the learning curves of the model trained for each subject. One can see that if the loss value for the validation dataset stops improving at some point and the loss value begin to increase, then it can be considered overfitting. Our learning curves, Figures 14 and 15, show that when training the model, the results do not suffer from overfitting. Furthermore, as mentioned in the previous Section 2.3, we also use the decoder network to reconstruct from the recurrence plot images to the sensor signal. After predicting the recurrence plot from the FNO model, we recover the sensor signal corresponding to the predicted recurrence plot image by using the decoder. Figure 16 shows the reconstructed sensor signal utilizing the trained decoder model after obtaining the predicted recurrence plot from the FNO model. We then plot some of the subjects' results and all the prediction results, but only a portion of the input is displayed. Since the prediction length of the FNO model is set to 20 time steps, the output of the decoder is also 20. However, for simplicity, we aggregate all prediction results except for duplicates and plot them at once, while the input signal plots only portions of them. Figure 16a shows the results for walking motion, indicating the x-, y-, and z-axis sequentially from the first row. The results show that despite image data type of the recurrence plot, the conversion results from the decoder work well. Similarly, the decoder's output for running motion in Figure 16b also shows that it has well recovered the sensor signal, and thanks to the satisfactory prediction results of the FNO model, it does not differ much from the actual target signals. Note that in this paper, the objective is constructing a personalized model for each subject by means of the modern deep learning model, the Fourier neural operator, and recurrence plots. The model presented in our experiments might be viewed as not generalized. However, we did not aim to make a generalized model that could predict the movement of various individuals with one model. A personalized model capable of predicting the movement of a specific person was considered in this paper. It is expected that this personalized model could be utilized not only to predict movements related to daily activities, but also to provide customized support such as exercise programs and physical therapy, which could lead to a wide range of health benefits.

	# 1	# 2	# 3	# 4	# 5	# 6
# 1	0.1577	1.3818	1.3234	2.2360	4.2557	2.4726
# 2	1.1809	0.1421	1.0700	0.6206	0.5160	0.6314
# 3	1.1302	1.0789	0.0809	0.5158	0.4672	0.6195
# 4	2.9517	2.5907	2.4724	0.1400	0.7720	1.0056
# 5	2.8226	2.5068	2.8779	1.1792	0.1608	1.2070
# 6	1.8792	1.5013	1.6142	0.6723	0.5520	0.0982

Figure 9. Mean squared error values of the FNO model for each subject for walking motions.

	# 1	# 2	# 3	# 4	# 5	# 6
# 1	0.0418	1.0851	1.7060	1.6427	1.7944	1.8008
# 2	1.0547	0.0716	1.4188	1.3892	1.0295	1.3629
# 3	0.8870	0.8038	0.1326	1.1889	1.4989	1.3575
# 4	1.6880	1.4914	2.2000	0.1352	0.6085	1.4244
# 5	2.6705	2.5907	3.4073	2.0169	0.1299	2.2370
# 6	1.1803	0.9667	1.5979	0.8955	0.5301	0.1050

Figure 10. Mean squared error values of the FNO model for each subject for running motions.

	# 1	# 2	# 3	# 4	# 5	# 6		# 1	# 2	# 3	# 4	# 5	# 6
# 1	0.2445	1.2144	1.3136	0.9590	0.9128	1.0294	# 1	0.4204	0.9351	1.4778	1.4418	0.9644	1.5282
# 2	1.1836	0.3141	0.8524	0.6783	0.6123	0.7444	# 2	0.9143	0.4720	1.3073	1.2911	0.8468	1.2399
# 3	1.2259	0.8318	0.3063	0.4960	0.5996	0.6985	# 3	0.8913	0.7868	0.5039	1.0168	1.0442	1.0801
# 4	2.4131	1.7426	1.4504	0.2248	0.6013	0.8236	# 4	1.7418	1.4292	1.6740	0.4509	0.6131	1.2310
# 5	2.8861	1.8125	1.7278	0.7963	0.3877	1.0050	# 5	2.4620	2.3670	2.9378	1.7902	0.4625	1.9818
# 6	2.1287	1.3539	1.3698	0.5888	0.5189	0.3564	# 6	1.2661	0.9006	1.2366	0.8170	0.5036	0.4164

(a)

(b)

Figure 11. MSE values of the CNN model for (a) walking motions and (b) for running motions.

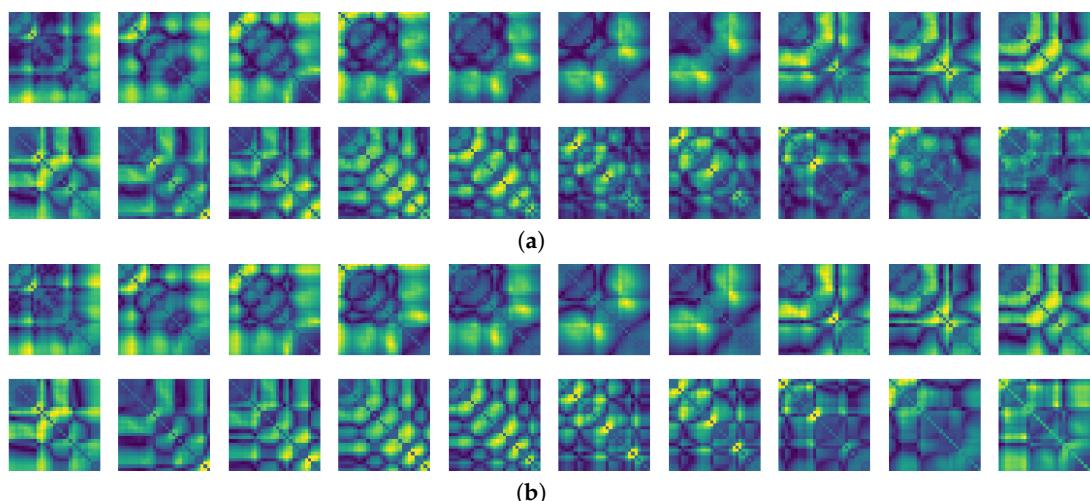


Figure 12. Cont.

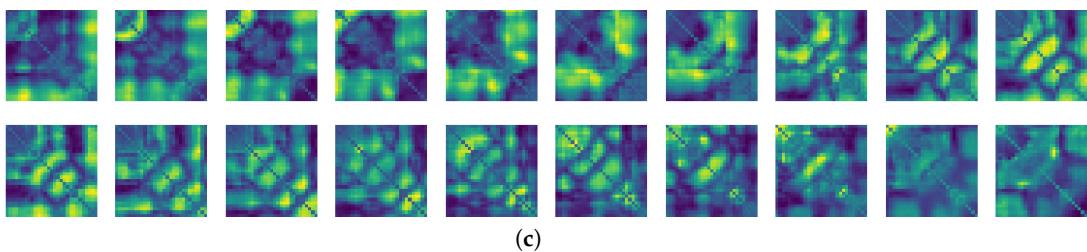


Figure 12. Some samples for walking motion: (a) predicted results of the FNO model, (b) target recurrence plot data, and (c) predicted results of the CNN model.

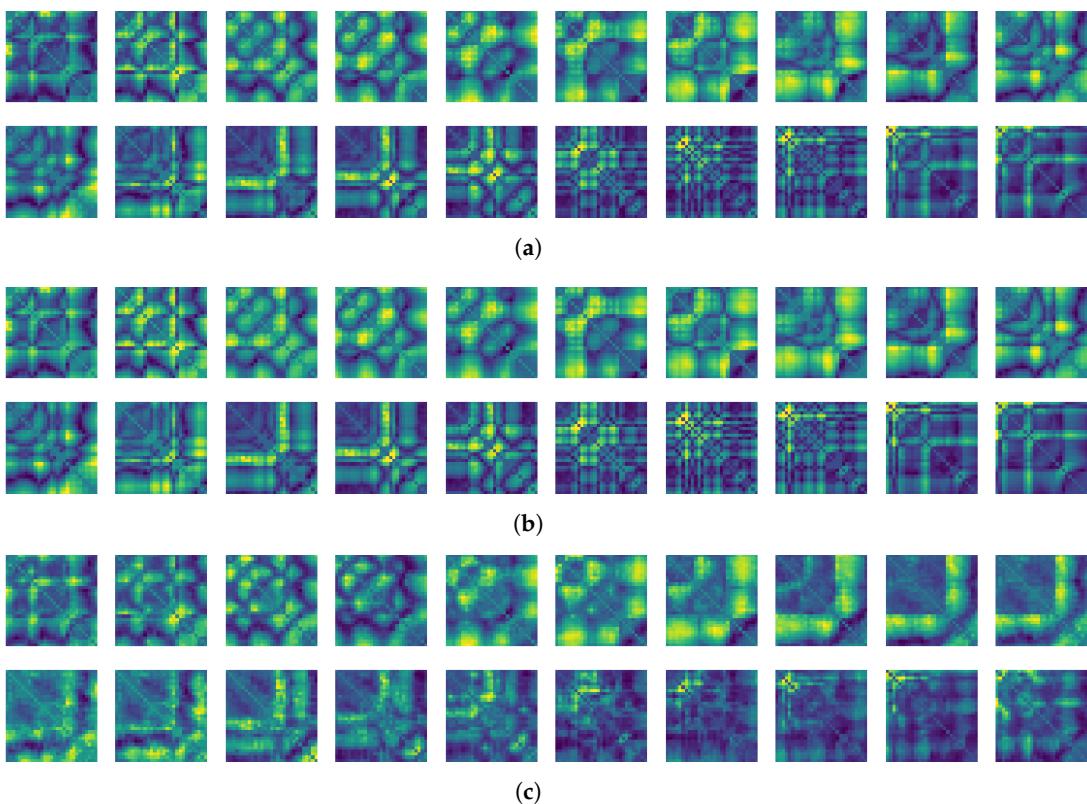
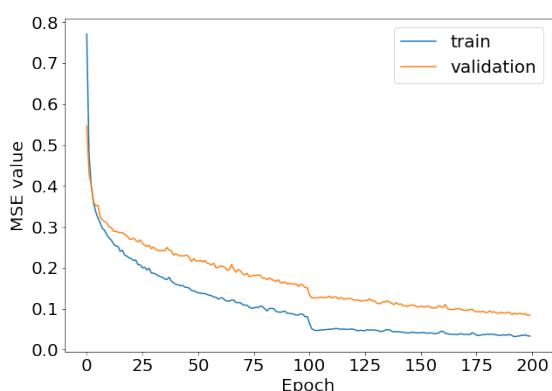
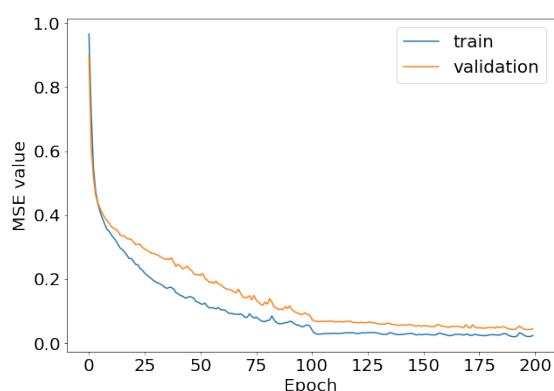


Figure 13. Some samples for running motion: (a) predicted results of the FNO model, (b) the target recurrence plot data, and (c) predicted results of the CNN model.



(a) One of the learning curves for walking.



(b) One of the learning curves for running.

Figure 14. Some of the learning curves of the trained FNO model for (a) walking and (b) running.

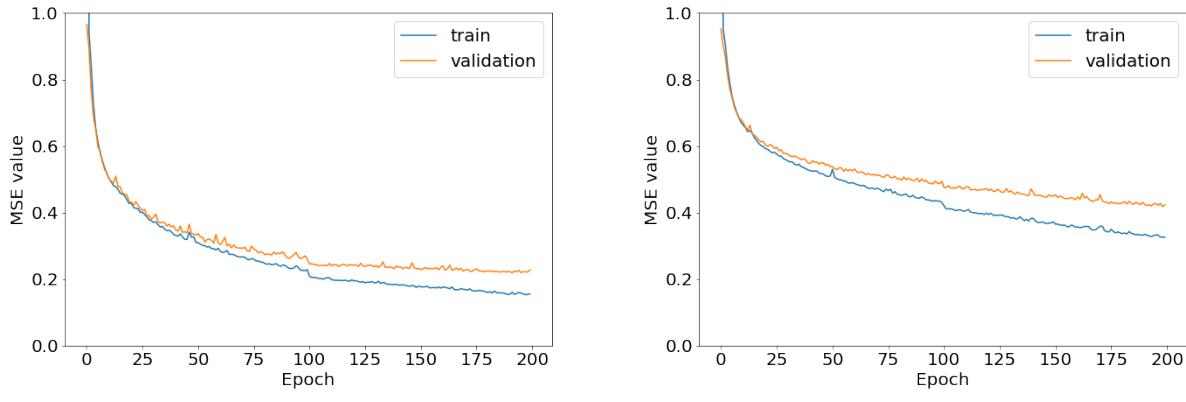


Figure 15. Some of the learning curves of the trained CNN model for (a) walking and (b) running.

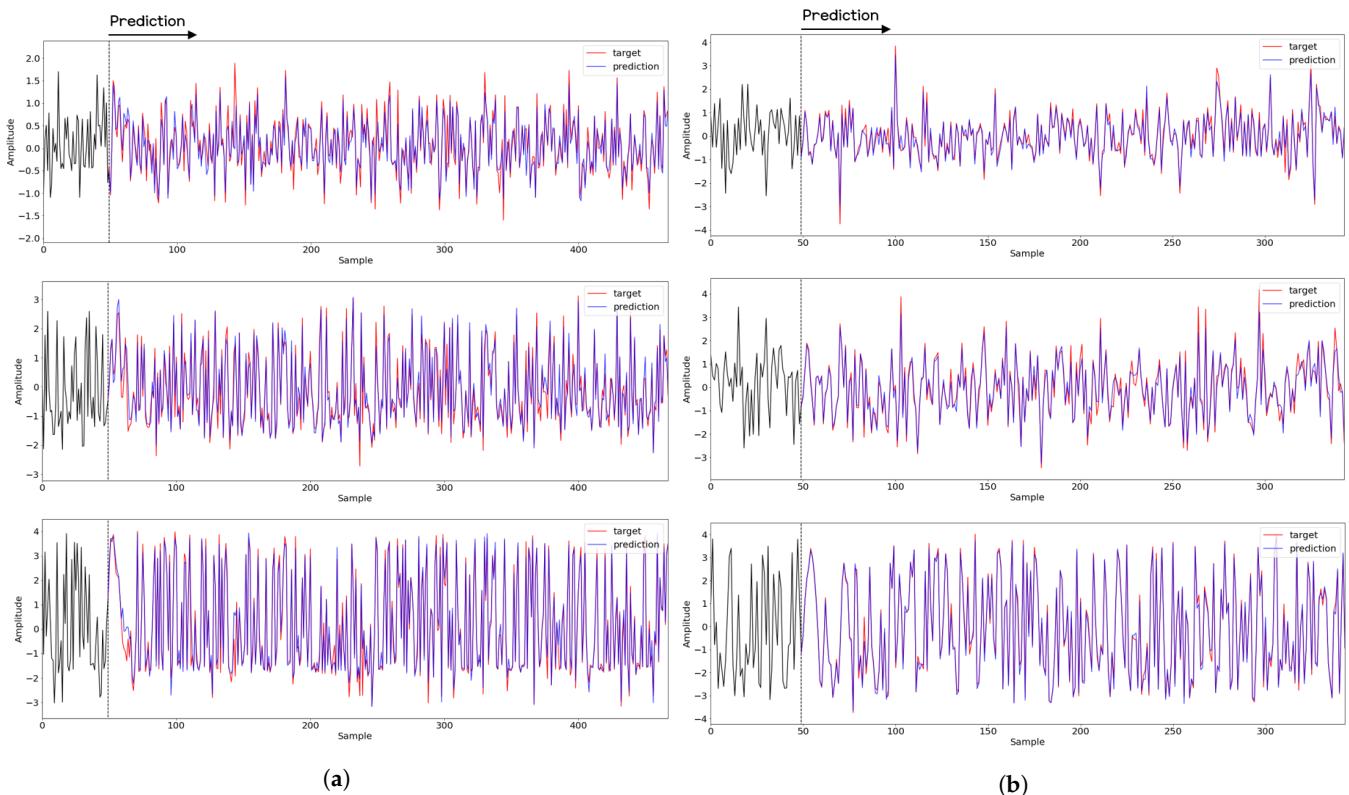


Figure 16. Recovery results for decoder network for (a) walking motion and (b) running motion. The top, middle, and bottom rows represent the x-, y-, and z-axis, respectively. The blue and red lines indicate the predicted signal from the FNO model and target signal, respectively, and the black line represents the signal corresponding to the input entered into the FNO model.

4. Discussion and Conclusions

4.1. Discussion

In this paper, a study was conducted to predict sensor signals for human motions (e.g., walking and running) using gyro sensors, recurrence plots, and a recent deep learning model, referred to as a Fourier neural operator (FNO). Our applications of FNO along with recurrence plots should be timely and meaningful. Activities such as walking and running are regular and somewhat periodic, but it is also true that they are indeed the most important components in daily life activities. Since methods utilizing frequency domain

information are generally efficient in dealing with periodic (or quasi-periodic) signals, we believe that the use of a Fourier neural operator, a modern innovative frequency domain approach, in characterizing daily activities merits investigation. Through the results of this paper, we show that when 30 samples, which are equal to 1 s of signals, were entered as input into the FNO model, prediction results were confirmed to be accurate for the next 20 samples, or about 0.6 s. The sensor signal is obtained from the gyroscope which is built in a smartphone, which is described in Section 3.1, and the data are converted into the recurrence plot to input into the FNO model, as described in Section 2.1. In the main experiment of this paper, we used only the gyroscope data. Smartphones contain various sensors, such as accelerometers, magnetometers, and gyroscopes, and we also consider using gyroscope and accelerometer data in this section. The procedure is the same as in the previous main experiment, but we consider replacing a section of the recurrence plot consisting of gyroscope data with data obtained from both the gyroscope and accelerometer. Referring to a previous study on creating recurrence plots from two sensors [52], the recurrence plot used in our main experiment is replaced by the cross recurrence plot, or CRP, as described in Section 2.1. Figures 17 and 18 show the MSE values for the test data for walking and running motions, respectively, when the FNO model was trained with the CRP. As seen in the results, even when the gyroscope and accelerometer data were used together, it was capable of creating personalized models for each subject. After models for each subject are trained with their own data, they show low MSE values when their own test data (i.e., data not used for training) are provided in the model, but show high MSE values when the test data of another subject is entered. On the other hand, the CNN model shows overall high MSE values (as seen in Figure 19), and it can be seen that the predictive model construction has failed as well as the personalized model. Not only the quantitative analysis of MSE values, but also the qualitative results from the cross recurrence plots predicted from the FNO model can be confirmed, as seen in Figures 20 and 21. Figure 20 is for walking motion, where the first row shows the predictive results of the FNO model, the second row shows the target cross recurrence plot that should be predicted from the FNO and CNN models, and the last row shows the outcomes predicted by the CNN model. These results indicate that the predicted cross recurrence plots by the FNO model are quite similar to the target, but those from the CNN model show differ from the target. Similarly, Figure 21 shows the predictive results of the FNO model for running motion, the target CRPs, and the predictive results of the CNN model. Similar to the walking motion results, this shows that the FNO model predicted the target well, but the CNN model did not. Figures 22 and 23 represent reconstructed sensor signals using the predicted CRP from the FNO model for walking and running motions, respectively. Since the gyroscope and the accelerometer are used in combination, the recovered sensor signals from the CRP also have two signals related to each sensor. In each figure, (a) represents the x, y, and z axes of the gyroscope in order from the top, and, similarly, (b) represents the x, y, and z axes of the accelerometer. Although it could be a rather difficult process to recover the signals from the CRP, in which information from the two sensors is combined, it can be seen that the recovered sensor signals via the decoder model are similar to the target signals.

	# 1	# 2	# 3	# 4	# 5	# 6
# 1	0.1181	0.7830	0.9405	0.6310	0.6311	0.6093
# 2	1.0696	0.0375	0.8034	0.5340	0.5356	0.5282
# 3	1.0337	0.6837	0.0659	0.3754	0.4731	0.4783
# 4	1.1189	0.7008	0.8142	0.0343	0.4372	0.4517
# 5	1.0750	0.6790	0.8600	0.4523	0.0439	0.4245
# 6	1.0665	0.6822	0.8410	0.4575	0.4525	0.0418

Figure 17. MSE values of the FNO model using CRP for each subject for walking motion.

	# 1	# 2	# 3	# 4	# 5	# 6
# 1	0.0482	0.8519	0.9024	0.8304	0.6870	0.9528
# 2	0.7240	0.0404	0.7706	0.7635	0.6661	0.8691
# 3	0.7258	0.7342	0.0452	0.7142	0.5845	0.8849
# 4	0.7061	0.7413	0.7421	0.0497	0.4846	0.7849
# 5	0.7571	0.7484	0.7858	0.5474	0.0540	0.8089
# 6	0.7189	0.7687	0.8310	0.5747	0.5899	0.0467

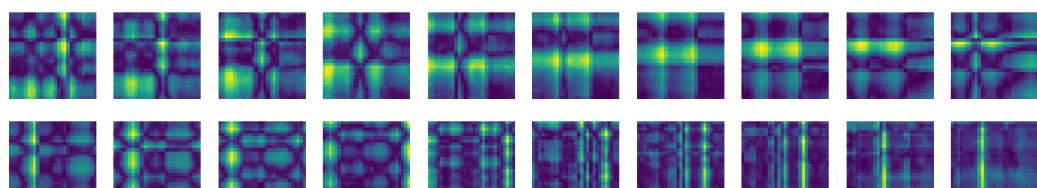
Figure 18. MSE values of the FNO model using CRP for each subject for running motion.

	# 1	# 2	# 3	# 4	# 5	# 6		# 1	# 2	# 3	# 4	# 5	# 6
# 1	0.9098	0.6667	0.8071	0.5489	0.5227	0.5519		0.3651	0.5390	0.7138	0.6908	0.6224	0.8341
# 2	0.9166	0.6425	0.8217	0.5119	0.4981	0.5173		0.5213	0.5896	0.6439	0.5431	0.4885	0.6962
# 3	1.0265	0.4531	0.4202	0.3879	0.4358	0.4547		0.5766	0.6489	0.6909	0.5603	0.5232	0.7381
# 4	0.9573	0.5993	0.6568	0.3816	0.4122	0.4152		0.6371	0.6964	0.6145	0.3965	0.3975	0.6645
# 5	0.9353	0.6474	0.8270	0.4990	0.4853	0.4928		0.5766	0.6391	0.7264	0.5726	0.4900	0.7698
# 6	1.0532	0.6438	0.7108	0.2799	0.3189	0.2973		0.5826	0.6279	0.6261	0.5007	0.4616	0.6405

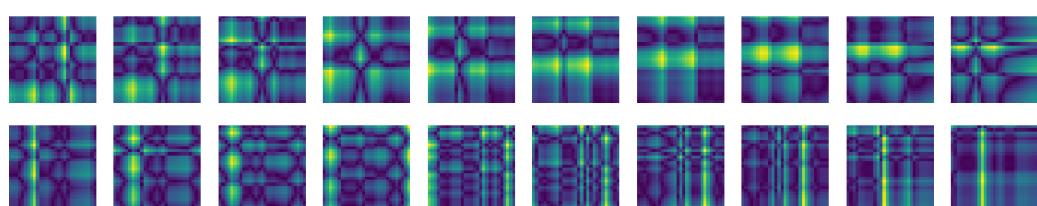
(a) Walking motion

(b) Running motion

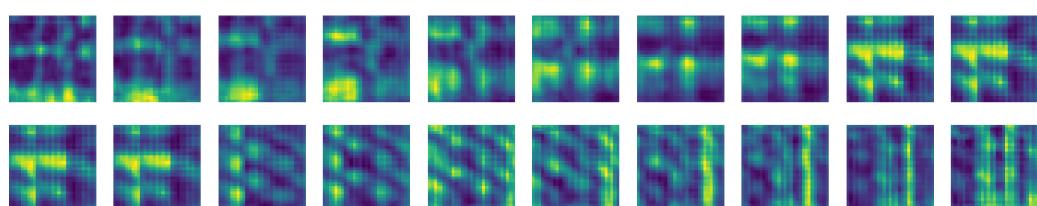
Figure 19. MSE values of the CNN model using CRP for each motion.



(a) Some samples predicted by the FNO model.

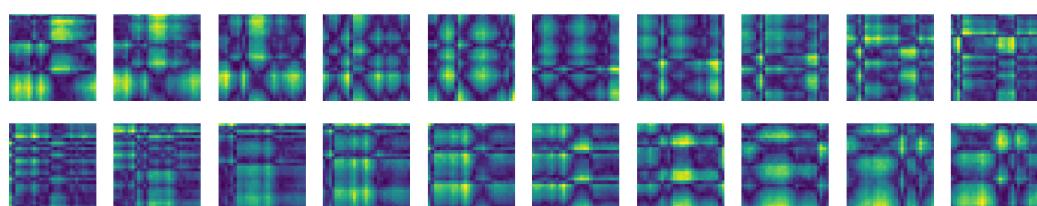


(b) The target cross recurrence plot to be predicted from the FNO model (a) and CNN model (c).

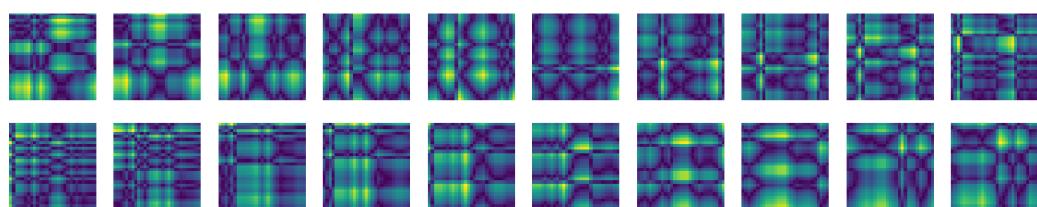


(c) Some samples predicted by the CNN model.

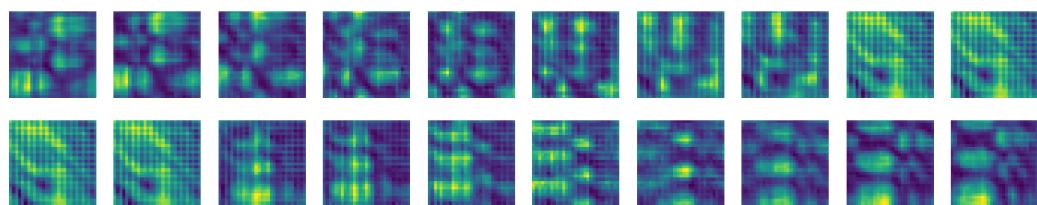
Figure 20. Some samples for walking motion: (a) predicted results of the FNO model, (b) target cross recurrence plot, and (c) predicted results of the CNN model.



(a) Some samples predicted by the FNO model.



(b) The target cross recurrence plot to be predicted from the FNO model (a) and CNN model (c).



(c) Some samples predicted by the CNN model.

Figure 21. Some samples for running motion: (a) predicted results of the FNO model, (b) target cross recurrence plot, and (c) predicted results of the CNN model.

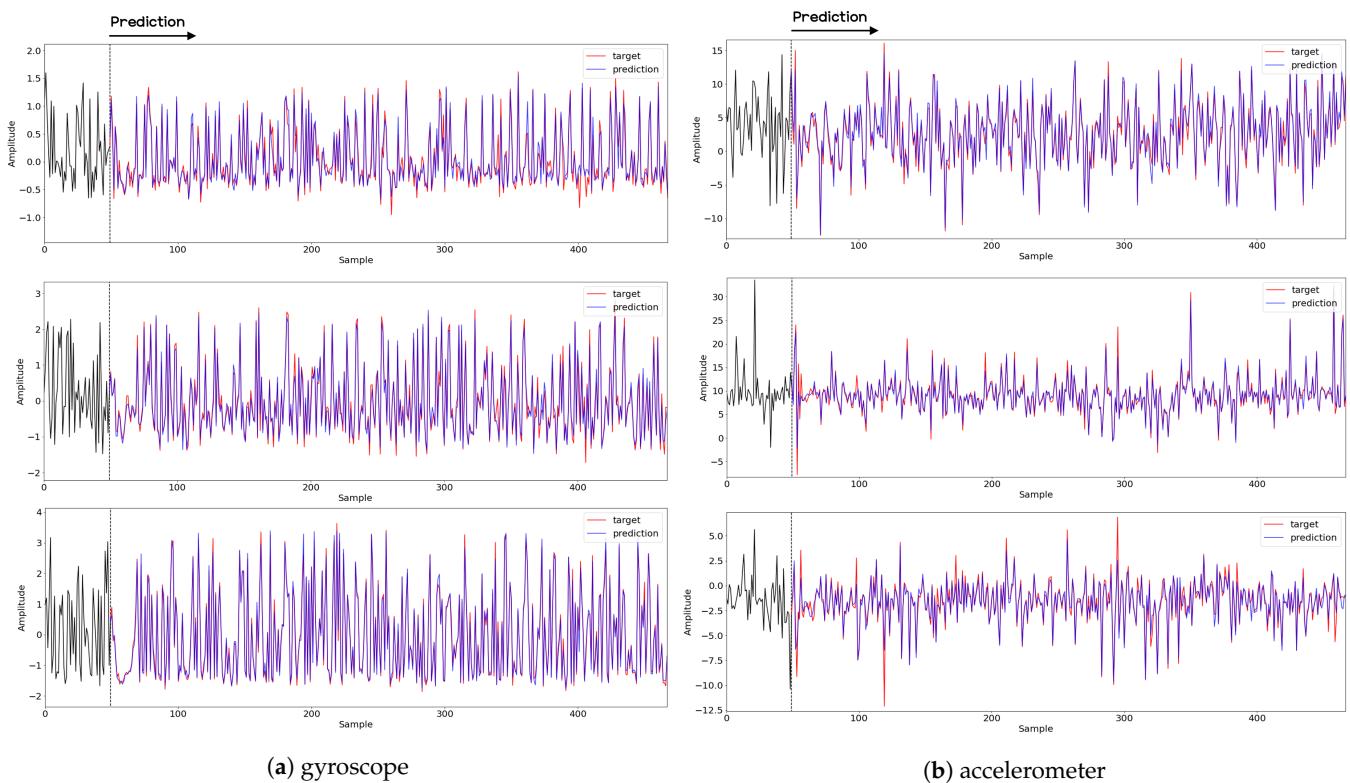


Figure 22. Samples of recovery results from the decoder model for walking motion. The top, middle, and bottom rows represent the x-, y-, and z-axis, respectively. The blue and red lines indicate the predicted signal from the FNO model and target signal, respectively, and the black line represents the signal corresponding to the input entered into the FNO model.

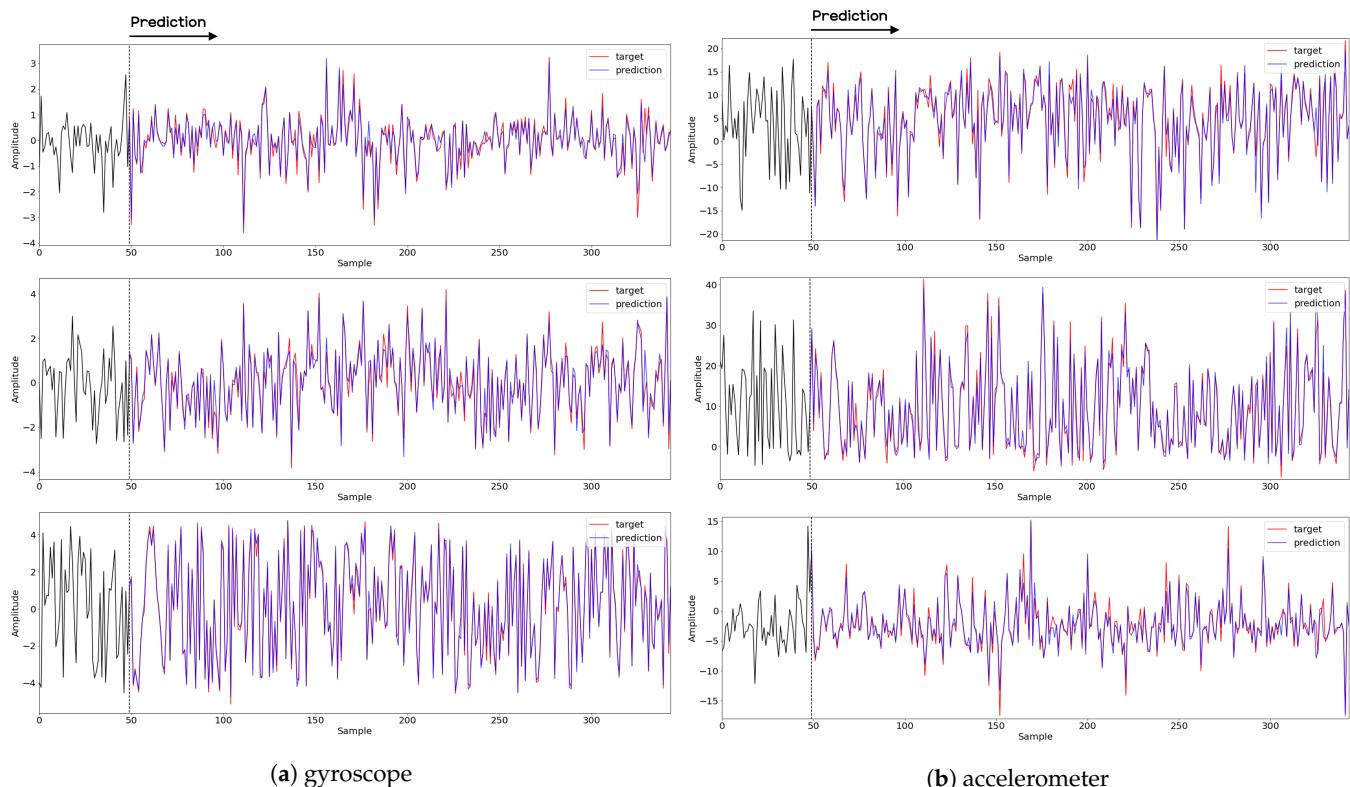


Figure 23. Samples of recovery results from the decoder model for running motion. The top, middle, and bottom rows represent the x-, y-, and z-axis, respectively. The blue and red lines indicate the predicted signal from the FNO model and target signal, respectively, and the black line represents the signal corresponding to the input entered into the FNO model.

We also considered walking up and down stairs, and for this experiment, we utilized a public dataset, MotionSense [41]. As mentioned in [41], the dataset was collected at 50 Hz, and an iPhone 6s was used as the smartphone device. The smartphone was located in the front trousers pocket. We conducted numerical experiments for the MotionSense dataset in a manner similar to our previous experiments. Figures 24 and 25 show the results of the FNO model and the CNN model, respectively. As shown in the figures, the MSE performance of the FNO model turns out to be better than for the CNN model, which shows that models trained on each individual subject can identify the subject.

One may have concerns that the presented approach lacks completeness in its human movement analysis. However, the main point of this paper is not to cover complete human movements, but to present a methodology for personalized healthcare. In this paper, we covered walking and running motions, which are the most common activities people perform in their daily lives, and this can be expanded to a variety of motions that comprise daily activities. Further study will be conducted, but we think that an expansion in scope would not be unreasonable because the FNO model also worked well for additional motions, such as walking up and down stairs. We expected that if further research like this is conducted, possible future applications would include personalized healthcare, such as assistance in physical exercise and rehabilitation therapy.

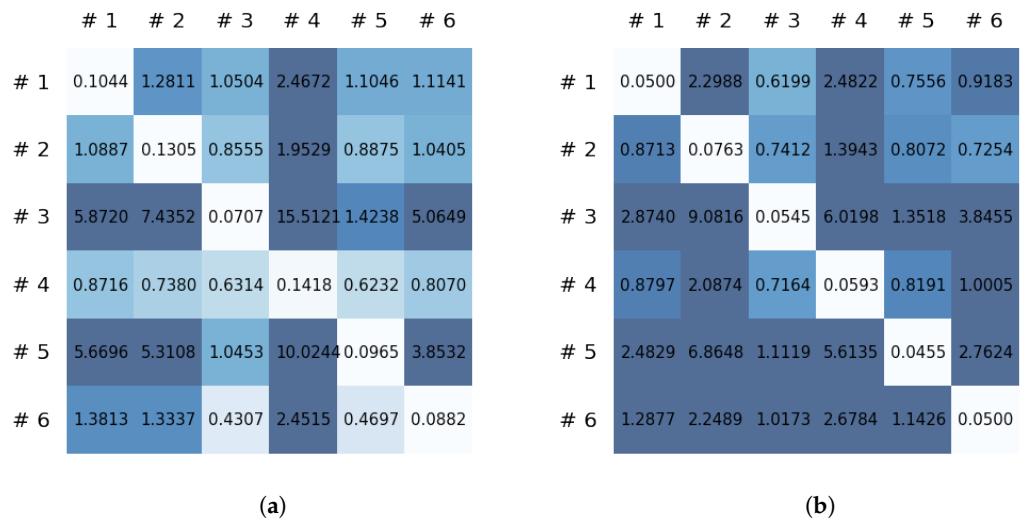


Figure 24. MSE values of the FNO model for (a) walking down stairs and (b) for walking up stairs.

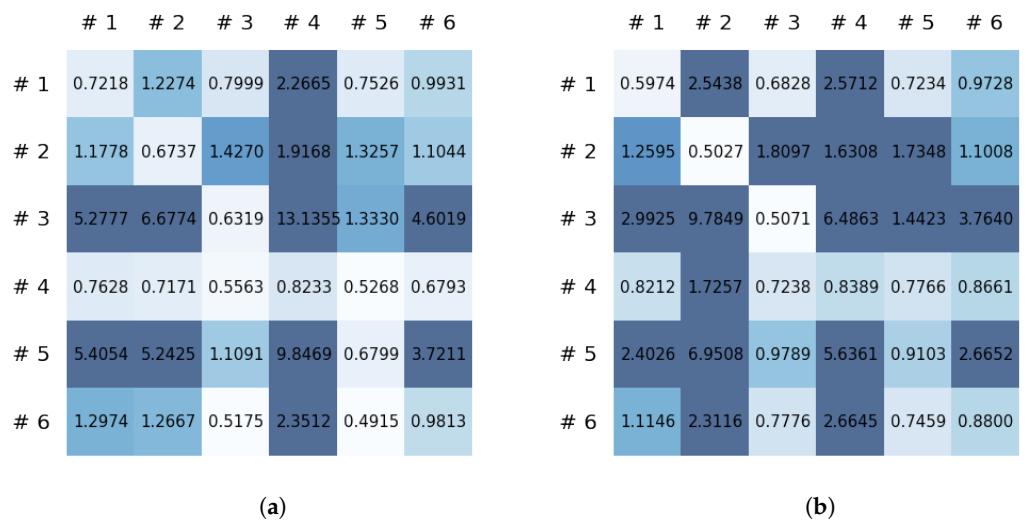


Figure 25. MSE values of the CNN model for (a) walking down stairs and (b) for walking up stairs.

Mesh Invariance. As discussed above and in previous research, FNO has the advantage of being mesh-independent. In order to confirm this characteristic of FNO, we first train the FNO model with the recurrence plot image, as introduced in Procedure 2, and the time window was set at 5. After training the FNO model, the trained weights were fixed and the time window corresponding to the image resolution was shortened. The results of the predicted recurrence plot image from varying the resolution of the recurrence plot are shown in Figure 26. As seen in the figure, the FNO model results are similar to the target recurrence plot images, but the results of the CNN model appear to differ from the target. From these results, we can experimentally check the mesh-free character of the Fourier neural operator [30].

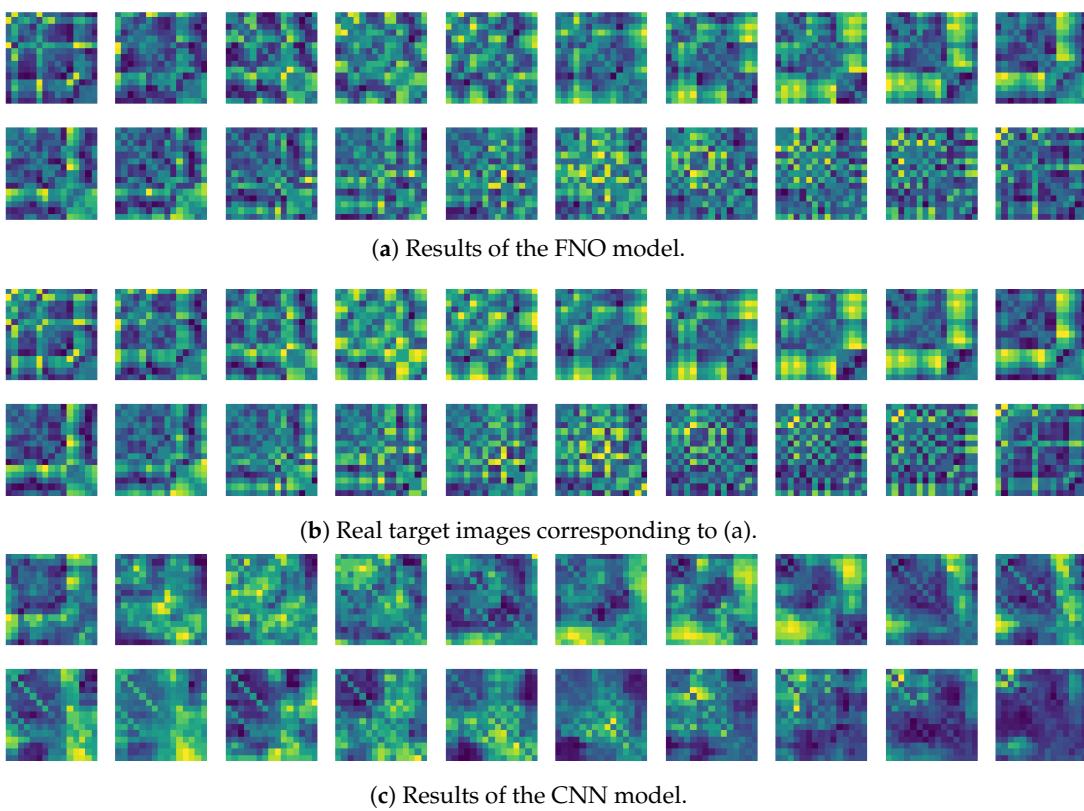


Figure 26. Some samples for running motion.

Fall Detection. According to the World Health Organization(WHO), falls are the second leading cause of death in the world, and are particularly fatal to adults over the age of 60 [53]. For these reasons, fall detection has become a major research field in the health area and the topic of many studies [54–56]. The results of this study confirm that human motion signals are well predicted using the FNO model and a gyro sensor. Accordingly, we consider fall detection by means of the FNO model and modern energy-based methods [57] for distinguishing abnormal movements corresponding to potential falls from normal motion. The energy-based method is particularly attractive, in that it does not require learning new parameters. Additionally, as introduced in one paper [57], the energy calculated from logit values makes for a clearer distinction between distributions of normal and abnormal movement than softmax, which is generally used for classification. Thus, if a model is trained to classify motions (e.g., walking and running), we can utilize it to detect falls without introducing or learning any new parameters. The problem of classifying activities of daily living (ADL) has been widely studied, with suitable performance demonstrated using deep learning methods [8,9]. One of the important capacities related to fall detection is to quickly alert guardians or supervisors when a fall occurs involving the elderly or infirm who are vulnerable to falls, and we consider the possible

usefulness of the proposed method for this capacity. Following the strategy of the one-class classification approach [58,59], we use the running and walking motion data as normal samples, and treat the motion data in the course of falling as abnormal samples. The model makes a decision about whether a given data sample is to be classified as normal, and when an anomaly occurs, we can quickly inform users of a risk of falling. In this paper, we constructed a simple deep neural network model composed of two fully connected layers, and trained it to classify input data such as walking or running. After training the classifying model, we use the energy computed from the logit values of the last layer and the threshold computed in the training phase to distinguish whether a behavior is normal (i.e., walking and running) or a likely fall. To utilize the successive prediction results of the FNO model, we fed the predicted time step recurrence plots as input into the classifier in the training and inference phase. It can thereby distinguish subsequent motion events predicted by the FNO model as a fall or as normal motion.

Falling conditions can thus be predicted more quickly than fall detection methods using current observational data. As shown in Figure 27, if a user is about to fall while walking, the FNO model can alert users of a falling risk slightly faster. When the classifier predicts the FNO model prediction results for the next time step as potential falling, it can provide the user an alert that the user may fall, and when actual falling data comes as an input after the falling risk signal, it can be classified and recorded as a fall. Through the results of this experiment on fall detection, it is possible to predict and notify users immediately before an actual fall occurs. Finally note that here, we only examined the potential of the proposed fall detection method for future applications. Therefore, it would be unsuitable for use in fall detection using the proposed method in its current form. In future work, we expect that this could be made more feasible for fall detection by acquiring data on various motions related to daily activities, which could expand the distribution of normal data, after conducting additional research.

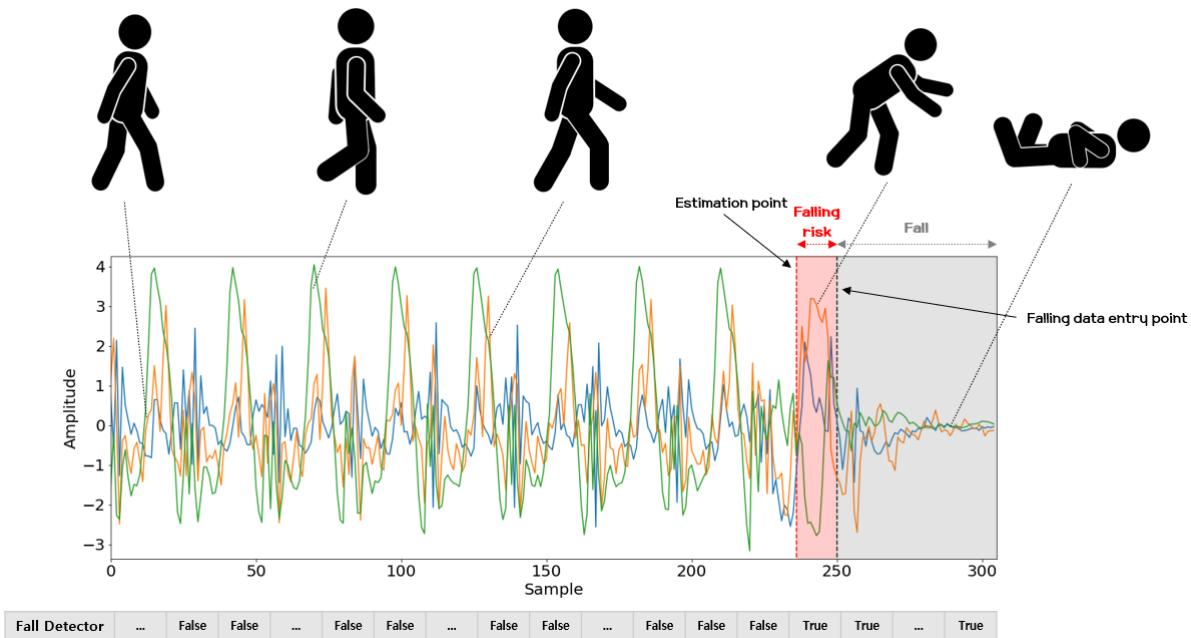


Figure 27. Results of fall detection experiments. The estimation point can be used as a point in time to transmit an indicator that informs the user in advance that a fall may occur when the fall detector detects a potential fall, by means of the FNO predictive results for the next time step data. After that, when actual fall data is entered, it can be viewed as a fall occurrence.

Gyroscope Drift. Since we use the gyroscope data in the experiments, there may be concern about drift, which is often an obvious challenge in long-term movements. In order to address this problem, an additional experiment was conducted on a subject. The subject performed the same walking motion as in the previous experiment in Section 3, and

performed it for 1 min after walking in place for 10 min at normal walking speed. Note that except for this difference (whereby the walking motion was performed after walking in place for 10 min), the experimental procedures and training conditions remain the same. Section (a) and (c) of Figure 28 show the recurrence plot prediction results for the FNO model and CNN model, respectively. As seen in these results, the FNO model turns out to have better predictive performance than the CNN model. In addition, Table 1 shows that the FNO model has lower MSE values compared to the CNN model. These observations indicate that the use of gyroscope data after walking in place for 10 min does not alter the results. Furthermore, we believe that the manner in which Equation (3) deals with gyroscope data, with the norm of relative difference, has less effect on the drift problem.

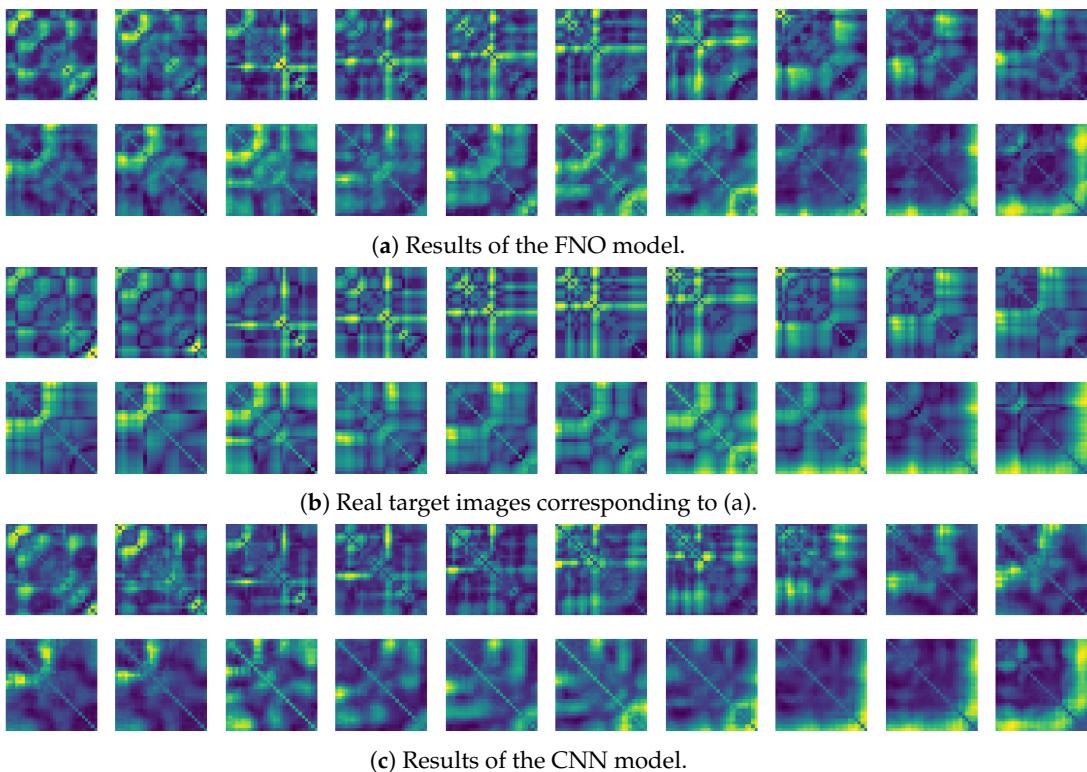


Figure 28. Some samples for walking motion using accelerometer.

Table 1. MSE values of the FNO model and CNN model for long-term walking motion.

Models	FNO	CNN
MSE value	0.134	0.332

4.2. Conclusions

In this paper, human motion prediction was performed using a gyro sensor and a modern deep learning method, the Fourier neural operator [30]. Predicting human motion has been being widely studied in many fields, with various studies on predicting human behavior in specific domains and environments, e.g., with surveillance cameras, as well as behavior prediction without space restrictions with wearable sensors. In this paper, a study is conducted to predict motion using a commonly used and popular device, specifically, a method using a gyro sensor built into a smartphone. The advantage of using wearable devices such as smartphones is that they can be used without spatial limitations or additional costs. For these reasons, a built-in smartphone gyro sensor and a modern deep learning method are used to analyze human motion signal input from the sensors to predict subsequent signals. Deep learning methods have recently demonstrated suitable performance, and among the various types of research being conducted, we utilize the

Fourier neural operator [30] for predicting the next time step of human behavior. In the results for the Fourier neural operator shown in Section 3.2, we checked the performance for predictive ability of the Fourier neural operator. As soon as sensor signal samples corresponding to 1 s are available, the next time step signal for the subsequent 0.6 s is predicted well by the Fourier neural operator. Furthermore, when the widely used convolutional neural network is used to construct and compare the same form for the FNO, we confirmed that the predictive performance for the Fourier neural operator is better than the predictive results from convolution networks. In Section 4.1, we also considered a method for detecting risks such as falling via human motion signals based on the predictive performance of the Fourier neural operator. Before the observed input signals arrive, the subsequent signal was predicted based on the Fourier neural operator, so it would be possible to warn users of potential danger slightly more quickly than when the observed signal is entered into the fall detector. It was thus confirmed that the subsequent signal can be predicted by means of pedestrian behavior signals obtained from a wearable sensor, and by means of the Fourier neural operator model. Furthermore, it was confirmed that when using the signal predicted by the Fourier neural operator, a warning of a fall risk can be provided slightly more quickly than by using only the current motion signal. In light of these findings, if wearable sensors and cloud servers can be used for this in the future, it will be possible to use these as a service to warn people in advance who might be seriously injured in a fall, such as the elderly.

Author Contributions: T.K. and J.P. (Jooyoung Park) conceived and designed the methodology of the paper; T.K. and J.P. (Jeongho Park) performed the experiments; T.K. wrote the computer program with help of J.L.; T.K. and J.P. (Jooyoung Park) wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2020R1F1A1072772) funded by the Korea government (MSIT).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of AIMATH Lab of Korea University (Project identification code: DL-2021-1, and date of approval: 1 July 2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2020R1F1A1072772) funded by the Korea government (MSIT).

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

References

- Ribeiro, P.M.S.; Matos, A.C.; Santos, P.H.; Cardoso, J.S. Machine Learning Improvements to Human Motion Tracking with IMUs. *Sensors* **2020**, *20*, 6383. [[CrossRef](#)]
- Wang, J.M.; Fleet, D.J.; Hertzmann, A. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 283–298. [[CrossRef](#)] [[PubMed](#)]
- Kim, T.; Park, J.; Heo, S.; Sung, K.; Park, J. Characterizing dynamic walking patterns and detecting falls with wearable sensors using Gaussian process methods. *Sensors* **2017**, *17*, 1172. [[CrossRef](#)] [[PubMed](#)]
- Kim, J.; Lee, J.; Jang, W.; Lee, S.; Kim, H.; Park, J. Two-stage latent dynamics modeling and filtering for characterizing individual walking and running patterns with smartphone sensors. *Sensors* **2019**, *19*, 2712. [[CrossRef](#)] [[PubMed](#)]
- Golestani, N.; Moghaddam, M. Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. *Nat. Commun.* **2020**, *11*, 1551. [[CrossRef](#)]
- Jordao, A.; Nazare, A.C., Jr.; Sena, J.; Schwartz, W.R. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. *arXiv* **2018**, arXiv:1806.05226.
- Zhang, Y.; Zhang, Y.; Zhang, Z.; Bao, J.; Song, Y. Human activity recognition based on time series analysis using U-Net. *arXiv* **2018**, arXiv:1809.08113.
- Wu, J.; Feng, Y.; Sun, P. Sensor fusion for recognition of activities of daily living. *Sensors* **2018**, *18*, 4029. [[CrossRef](#)] [[PubMed](#)]

9. Wang, A.; Zhao, S.; Zheng, C.; Yang, J.; Chen, G.; Chang, C.Y. Activities of Daily Living Recognition With Binary Environment Sensors Using Deep Learning: A Comparative Study. *IEEE Sens. J.* **2020**, *21*, 5423–5433. [[CrossRef](#)]
10. Cao, Z.; Gao, H.; Mangalam, K.; Cai, Q.Z.; Vo, M.; Malik, J. Long-term human motion prediction with scene context. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 387–404.
11. Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; Zisserman, A. Counting out time: Class agnostic video repetition counting in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10387–10396.
12. Muro-De-La-Herran, A.; Garcia-Zapirain, B.; Mendez-Zorrilla, A. Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications. *Sensors* **2014**, *14*, 3362–3394. [[CrossRef](#)]
13. Nguyen, M.D.; Mun, K.R.; Jung, D.; Han, J.; Park, M.; Kim, J.; Kim, J. IMU-based spectrogram approach with deep convolutional neural networks for gait classification. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 4–6 January 2020; pp. 1–6.
14. Alsheikh, M.A.; Selim, A.; Niyato, D.; Doyle, L.; Lin, S.; Tan, H.P. Deep activity recognition models with triaxial accelerometers. In Proceedings of the Workshops at the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–13 February 2016.
15. Jia, Y.; Song, R.; Wang, G.; Yan, C.; Guo, Y.; Zhong, X. Human Activity Classification with Multi-frequency Spectrogram Fusion and Deep Learning. In Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), Wuxi, China, 19–21 July 2019; pp. 117–121.
16. Liu, Z.; Xu, L.; Jia, Y.; Guo, S. Human Activity Recognition Based on Deep Learning with Multi-spectrogram. In Proceedings of the 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 23–25 October 2020; pp. 11–15.
17. Zheng, X.; Wang, M.; Ordieres-Meré, J. Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0. *Sensors* **2018**, *18*, 2146. [[CrossRef](#)] [[PubMed](#)]
18. Hur, T.; Bang, J.; Lee, J.; Kim, J.I.; Lee, S. Iss2Image: A novel signal-encoding technique for CNN-based human activity recognition. *Sensors* **2018**, *18*, 3910. [[CrossRef](#)] [[PubMed](#)]
19. Lu, J.; Tong, K.Y. Robust single accelerometer-based activity recognition using modified recurrence plot. *IEEE Sens. J.* **2019**, *19*, 6317–6324. [[CrossRef](#)]
20. Garcia-Ceja, E.; Uddin, M.Z.; Torresen, J. Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition. *Procedia Comput. Sci.* **2018**, *130*, 157–163. [[CrossRef](#)]
21. Jianjie, L.; Raymond, T. Encoding Accelerometer Signals as Images for Activity Recognition Using Residual Neural Network. Available online: <https://arxiv.org/vc/arxiv/papers/1803.1803.09052v1.pdf> (accessed on 1 March 2018).
22. Penatti, O.A.; Santos, M.F. Human activity recognition from mobile inertial sensors using recurrence plots. *arXiv* **2017**, arXiv:1712.01429.
23. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
24. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
25. Oord, A.V.D.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv* **2016**, arXiv:1609.03499.
26. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 213–229.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
28. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Multipole graph neural operator for parametric partial differential equations. *arXiv* **2020**, arXiv:2006.09535.
29. Anandkumar, A.; Azizzadenesheli, K.; Bhattacharya, K.; Kovachki, N.; Li, Z.; Liu, B.; Stuart, A. Neural Operator: Graph Kernel Network for Partial Differential Equations. In Proceedings of the ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations, Virtual Conference, Addis Ababa, Ethiopia, 26 April 2020.
30. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; An kumar, A. Fourier neural operator for parametric partial differential equations. *arXiv* **2020**, arXiv:2010.08895.
31. Chen, R.T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural ordinary differential equations. *arXiv* **2018**, arXiv:1806.07366.
32. Hatami, N.; Gavet, Y.; Debayle, J. Classification of time-series images using deep convolutional neural networks. In Proceedings of the Tenth International Conference on Machine Vision (ICMV 2017), Vienna, Austria, 13–15 November 2017; International Society for Optics and Photonics: Bellingham, WA, USA, 2018; Volume 10696, p. 106960Y.
33. Zhang, Y.; Hou, Y.; Zhou, S.; Ouyang, K. Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks. *Sensors* **2020**, *20*, 3818. [[CrossRef](#)]
34. Eckmann, J.P.; Kamphorst, S.O.; Ruelle, D. Recurrence plots of dynamical systems. *World Sci. Ser. Nonlinear Sci. Ser. A* **1995**, *16*, 441–446.

35. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
36. Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; Ontanon, S. FNet: Mixing Tokens with Fourier Transforms. *arXiv* **2021**, arXiv:2105.03824.
37. Lu, L.; Jin, P.; Karniadakis, G.E. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv* **2019**, arXiv:1910.03193.
38. Guiry, J.J.; van de Ven, P.; Nelson, J.; Warmerdam, L.; Riper, H. Activity recognition with smartphone support. *Med. Eng. Phys.* **2014**, *36*, 670–675. [[CrossRef](#)]
39. Micucci, D.; Mobilio, M.; Napoletano, P. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Appl. Sci.* **2017**, *7*, 1101. [[CrossRef](#)]
40. Huang, E.J.; Onnela, J.P. Augmented Movelet Method for Activity Classification Using Smartphone Gyroscope and Accelerometer Data. *Sensors* **2020**, *20*, 3706. [[CrossRef](#)] [[PubMed](#)]
41. Malekzadeh, M.; Clegg, R.G.; Cavallaro, A.; Haddadi, H. Mobile sensor data anonymization. In Proceedings of the International Conference on Internet of Things Design and Implementation, Montreal, QC, Canada, 15–18 April 2019; pp. 49–58.
42. Vavoulas, G.; Chatzaki, C.; Malliotakis, T.; Pediaditis, M.; Tsiknakis, M. The mobiact dataset: Recognition of activities of daily living using smartphones. In Proceedings of the International Conference on Information and Communication Technologies for Ageing Well and e-Health, Rome, Italy, 21–22 April 2016; Volume 2, pp. 143–151.
43. Matlab Application. Available online: <https://apps.apple.com/us/app/matlab-mobile/id370976661> (accessed on 9 December 2021).
44. iPhone XS Specification. Available online: https://support.apple.com/kb/SP779?locale=en_US (accessed on 12 September 2018).
45. Shoaib, M.; Bosch, S.; Incel, O.D.; Scholten, H.; Havinga, P.J. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* **2014**, *14*, 10146–10176. [[CrossRef](#)]
46. Shoaib, M.; Scholten, H.; Havinga, P.J. Towards physical activity recognition using smartphone sensors. In Proceedings of the 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, Vietri sul Mare, Italy, 18–21 December 2013; pp. 80–87.
47. San Buenaventura, C.V.; Tiglao, N.M.C. Basic human activity recognition based on sensor fusion in smartphones. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; pp. 1182–1185.
48. Sousa Lima, W.; Souto, E.; El-Khatib, K.; Jalali, R.; Gama, J. Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors* **2019**, *19*, 3213. [[CrossRef](#)] [[PubMed](#)]
49. Definition of Walking and Running, Walk Jog Run Club. Available online: <http://www.wjrclub.com/terms-and-definitions.html> (accessed on 21 September 2001).
50. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberl, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [[CrossRef](#)]
51. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
52. DelMarco, S.; Deng, Y. Detection of chaotic dynamics in human gait signals from mobile devices. In Proceedings of the Mobile Multimedia/Image Processing, Security, and Applications, Anaheim, CA, USA, 22 June 2017; International Society for Optics and Photonics: Bellingham, WA, USA; Volume 10221, p. 102210C.
53. World Health Organization. Falls. Available online: <https://www.who.int/news-room/fact-sheets/detail/falls> (accessed on 26 April 2021).
54. Luna-Perejón, F.; Domínguez-Morales, M.J.; Civit-Balcells, A. Wearable fall detector using recurrent neural networks. *Sensors* **2019**, *19*, 4885. [[CrossRef](#)] [[PubMed](#)]
55. Santos, G.L.; Endo, P.T.; Monteiro, K.H.D.C.; Rocha, E.D.S.; Silva, I.; Lynn, T. Accelerometer-based human fall detection using convolutional neural networks. *Sensors* **2019**, *19*, 1644. [[CrossRef](#)]
56. Gutiérrez, J.; Rodríguez, V.; Martín, S. Comprehensive review of vision-based fall detection systems. *Sensors* **2021**, *21*, 947. [[CrossRef](#)]
57. Liu, W.; Wang, X.; Owens, J.D.; Li, Y. Energy-based Out-of-distribution Detection. *arXiv* **2020**, arXiv:2010.03759.
58. Zhang, T.; Wang, J.; Xu, L.; Liu, P. Fall detection by wearable sensor and one-class SVM algorithm. In Proceedings of the Intelligent Computing in Signal Processing and Pattern Recognition, Kunming, China, 16–19 August 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 858–863.
59. Santoyo-Ramón, J.A.; Casilar, E.; Cano-García, J.M. A study of one-class classification algorithms for wearable fall sensors. *Biosensors* **2021**, *11*, 284. [[CrossRef](#)]