Human identification method based on motion sensors in wearable devices

Zihao Ding
Rutgers University
New Brunswick, US
zd75@scarletmail.rutgers.edu

Hongpeng Zhang
Rutgers University
New Brunswick, US
hz227@scarletmail.rutgers.edu

ABSTRACT

Identification based on motion features is a leading technology of biometric recognition. Compared to other popular identification methods such as fingerprints or face recognition, motion feature identification (MFI) has unique advantages that make it suitable for IoT applications. First, the sources of motion data are vast as nearly every type of wearable or portable device has motion sensors like accelerometers and gyroscopes installed. Therefore, MFI can be applied in any situation without specific, expensive equipment. Second, motion data is more respectful to the concern of privacy matters than some other forms of identification. Unlike a picture of a human face or a fingerprint, motion data has complex structures that are difficult for humans to comprehend; leak of such data will not cause major damage to the victim's privacy. Considering these pros, we build a method of extracting motion features from motion sensor data from wearable devices for identification. We set out optimize the data processing and recognition methods from existing studies on identification based on motion features. In this paper, we modified the raw accelerometer data and used time-frequency distribution to capture the pattern of an individual's motion. Then, we built a classification method based on Visual Geometry Group (VGG) transfer learning. We used VGG transfer learning due to the size limitations on our dataset and the available computing power. Finally, we analyzed our result and proposed improvements to the project.

KEYWORDS

motion feature identification; wearable devices; accelerometer; time frequency representation; CNN

1 Introduction

The goal of motion feature identification (MFI) is to use the accelerometer data to identify people. Different from iris [1], face [2], or fingerprint [3] recognition, motion feature identification is a non-invasive, non-intrusive method; it does not require the person to cooperate or behave in a certain way. No personal sensitive data is collected during MFI, and it is hard for person to disguise his or her behavior (unless there is physical change such as carrying a heavy load or a different type of shoe [4]). In recent years, there have been multiple approaches to motion feature identification; most of them rely on computer vision due to ease of access and deployment, especially in video surveillance [5]. However, with the spread of wearable devices like smartwatches and activity trackers as well as the popularity of mobile phones, it has become possible to capture the dynamics of a person's movement in different dimensions [6]. Sensor-based motion feature identification avoids some problems in computer vision-based approaches, but it has its own limitations. Most wearable devices use Micro-Electro-Mechanical-System (MEMS) sensors. Accelerometers are one of the most common electromechanical devices in the MEMS sensors. Accelerometers can measure acceleration force in different axes with a high sampling rate, making them the perfect, economical approach to MFI.

In this paper, we build an adaptive motion feature identification solution based on CNN. Our project is divided into the following parts. First, we use real-world time series data collected from the motion sensors of wearable devices on various positions of the human body. Since the motion signal has multiple components in frequency and time-space, we then formulate the long-sized time series data, turning it into a 2-D map. Then, we map the input signal to the time and frequency space by applying an appropriate time-frequency distribution (TFD). After that, we construct a convolutional neural network (CNN) to process the 2-D map image we generated from the sensor's time series data to identify people. To verify our result, we divided the 2-D map image into three parts to train, validate, and test the model. Finally, we compare the results from different TFD methods and determine which TFD method is better for human identification. Then, we analyze what we can do to further improve the results.

The organization of this paper is as follows. Section 1 is the introduction. Section 2 of the paper introduces the related studies. Section 3 presents the problems and challenges we face in this project. Section 4 discusses in detail the methods we used and how they worked. Section 5 provides the technical details about the experiment. Section 6 presents the results of our experiment. In Section 7, we list our future works. Lastly, Section 8 is the conclusion.

2 Related Works

Using accelerometers in wearable devices as MFI inputs has its challenges. Since wearable devices are supposed to be worn or placed wherever the user wants, the relative orientation between the accelerometer and the person's body will hardly be the same between different sessions [7]. Since accelerometers measure acceleration force in their preset axes, any rotation of the placement of the sensor will change the results of measurement. To overcome this issue, we followed [8] approach to solve the orientation dependency issue. Separating motion data into reasonable cycles for processing is another problem we encountered. The simple peak to peak method is inaccurate due to the complexity of motion.

There has recently been much research done in the MFI field. For the identification part, this work mainly separates into two approaches: the similarity measure approach and the machine learning approach. For the similarity approach, some examples include Liu [9], who used a dynamic time warping (DTW) algorithm for matching data, and Subramanian [10], who used Tanimoto similarity measurement. The most recent works took the machine learning approach. Various modeling algorithms such as decision trees [12], linear discriminant analysis (LDA) [13], support-vector machines (SVM) [13], Bayesian [14], and the k-nearest-neighbors algorithm (k-NN) [14] have been used in recent studies of motion feature identification. However, the performance of machine learning MFI systems are highly dependent on data preprocessing and feature extraction. The signal quality of the sensors as well as voluntary or involuntary user movements increase the difficulty of extracting relevant features from the data. In this case, manual feature extraction, as seen in [15], to process data used for machine learning systems is subjective, time consuming, and might lead to biased results.

In our paper, we demonstrate a different approach to MFI using motion cycle extraction, data visualization, and CNN-based classifiers instead of traditional time series analysis. We then compared the prediction results based on the two data visualization techniques we used.

3 Problems and Challenges

As mentioned in the previous section, the main goal of this paper is to develop a system that can identify people based on any kind of accelerometer data. The input of our system is accelerometer data. Its output is a confusion matrix showing the identification result. To build this system, there are several problems we must solve. Since we decided to use CNN as the classifier to identify people, preprocessing data is a challenge. We need to find a way to extract enough features to build a 2-D map image to feed the CNN. The orientation between the accelerometer and the individual's body is also unknown. We must develop a universal solution to process the data from the accelerometer's various axes. After all, the pre-processing of raw data depending on the final size of our data may not be sufficient to train a CNN model that is accurate enough to identify people from scratch.

4 Methodology

We aim to recognize the target's identity solely using the data of human movement collected by IMU devices worn by people. To achieve this goal, we must extract and identify characteristics hidden inside the lengthy time-series data the basic premise of this project.

The overview of our project is shown in Figure 1. The method we use consists of three parts. The first part is about data collection: the volunteers wear several IMU devices on various parts of their body and repeatedly make the same movement for a while so the raw data in time domain can be collected. Then comes data processing and visualization. In this part, we select data series from certain IMUs to extract cycle data after preprocessing. We believe a considerable number of daily movements are periodic, so the motion features contained in these cycles can be detected through comparisons among these cycles. In the last step, we use a convolutional neural network (CNN) to detect and identify motion features. CNN is famous and useful in the field of computer vision. Since we transformed the feature detection task to a visual mission by visualization in Step 2, we can use this powerful tool to solve our problem.

4.1 Data Collection

Our experiment required IMU data from the various body part of different people during diverse types of motion. Due to the limitation of hardware, time, and space—especially during the COVID-19 pandemic—we sought out existing third-party datasets which fulfilled our requirements. The dataset we used for this paper is from DataSet RealWorld (HAR), created by the University of Mannheim Research Group Data and Web Science. It includes acceleration, GPS, gyroscope, light, magnetic field, and sound level

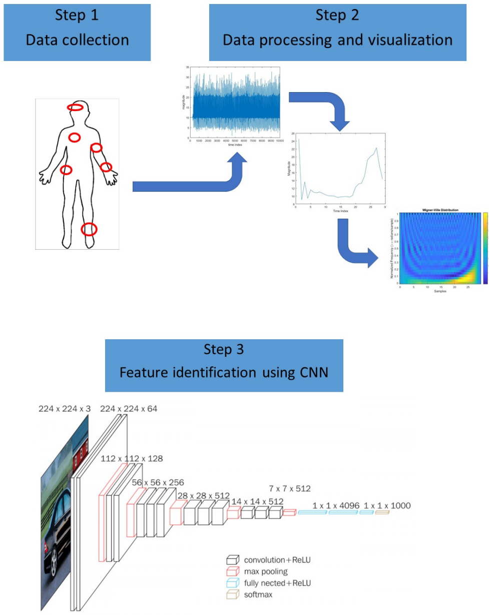


Fig 1. The overview structure of the whole identification method, including data collection from wearable IMU, extract and visualize motion cycle from the time-series raw data, and use CNN to detect and identify motion features.

data from fifteen people (age 31.9±12.4 years, height 173.1±6.9cm, weight 74.1±13.8kg, eight males and seven females) [18]. The participants performed several activities, including climbing up and down, jumping, lying, standing, sitting, running/jogging, and walking. Each person performed each activity for about 10 minutes, except for jumping due to the physical exertion (~1.7 minutes). For each activity, they simultaneously recorded the acceleration of the chest, forearm, head, shin, thigh, upper arm, and waist.

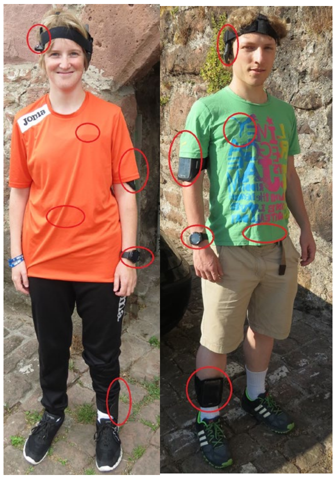
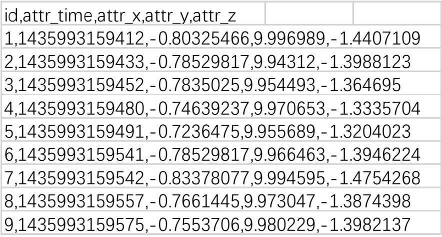


Fig 2. Two volunteers of the HAR dataset; the red circles indicate the sensor locations.

The HAR dataset contains much more detail than we needed. We chose to focus on the acceleration data collected by the shin-mounted sensors of ten people during their walking because walking is one of the most basic motions in people's daily lives. Therefore, it is data-rich. Additionally, a shin-mounted accelerometer can detect the strongest periodic signals during walking among those locations.

4.2 Data Pre-Processing

The acceleration data we need is stored as time-series data. One of the data series is shown on Figure 3. Such data series include 20,000 data points, with each point's time index, and acceleration values along three axes.



```
id,attr_time,attr_x,attr_y,attr_z
1,1435993159412,-0.80325466,9.996989,-1.4407109
2,1435993159433,-0.78529817,9.94312,-1.3988123
3,1435993159452,-0.7835025,9.954493,-1.364695
4,1435993159480,-0.74639237,9.970653,-1.3335704
5,1435993159491,-0.7236475,9.955689,-1.3204023
6,1435993159541,-0.78529817,9.966463,-1.3946224
7,1435993159542,-0.83378077,9.994595,-1.4754268
8,1435993159557,-0.7661445,9.973047,-1.3874398
9,1435993159575,-0.7553706,9.980229,-1.3982137
```

Fig 3. Part of the acceleration data stored in .csv files

However, we should not directly use these 3-axis data because it is too crowded and contains orientation-related bias. The orientation-related bias is caused by subtle orientation differences devices worn by different people. Such differences can affect the 3-axis data and become a major source of noise. To reduce such noise, we use the magnitude of the acceleration instead of the original 3-axis values. The magnitude can be calculated through the vector sum, which is shown in Equation 1:

$$Mag(t) = \sqrt{R_x^2 + R_y^2 + R_z^2} \tag{1}$$

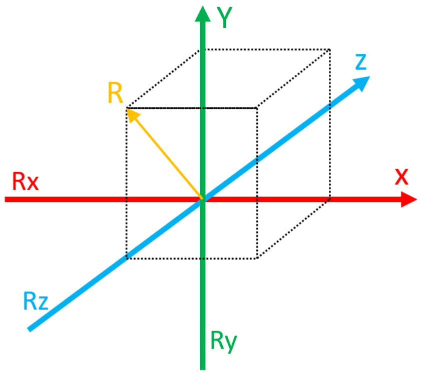


Fig 4. Graphical illustration of Factor Extraction. R is the Result factor.

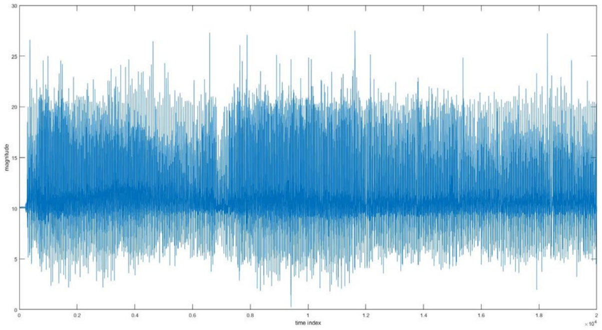


Fig 5. One data series after calculating magnitude

4.3 Cycle extraction

After pre-processing, the dataset was still with 20,000 data points. For the single long time series data, this size makes it hard to extract features. Additionally, the length of data can influence feature detection; as the differences in length cause different characteristics in the time series data. In reality, it is impossible to collect the exact same signal lengths for an experiment. As a result, we must normalize the time series data to make different data series have a similar length. This goal can be achieved by cycle extraction. The idea behind it is simple: most motions, like walking, are periodic. Every step moved by one person is similar to that person's other steps. If we can analyze several of those motion cycles, we may be able to extract the motion features by "summarizing" them. Therefore, we must divide the whole series into cycles.

To do this, we must first know the cycle frequency. Otherwise, we will tear features apart. We used the frequency-domain analysis to get the motion frequency from the time-domain signal. In other words, we transformed the signal to the frequency domain by means of the Fourier transformation:

$$X(w) = \int x(t)e^{-jt\omega}\,dt \tag{2}$$

The Fourier transformation reveals the frequency component of the signal. We must to choose one of those components as the cycle frequency. As the cycle frequency is caused by the actual frequency of the physical movement, there are two features that cycle frequency must satisfy. First, it must be a dominant frequency which is a peak on the frequency-domain signal because it is caused by the dominant movement. Second, it must be reasonable in reality. For example, most people walk with a frequency between 0.5-2Hz, so the cycle frequency should also lie in this interval. Figure 6 shows part of the Fourier transformation of the signal in Figure 5.

The second step of cycle extraction is to get the cycle length in terms of the time index. This is needed to divide the signal. The cycle length can be calculated through Equation 3. The single cycle extracted is shown in Figure 6.

$$L_{cycle} = \frac{f_{sampling}}{f_{cycle}} \tag{3}$$

Note that the cycles we extracted are not the exact physical cycles of human motions. For example, the starting point of each extracted cycle may not be the beginning of the physical step. For each subject, each section of his/her cycle data may not look similar to other sections. This is because we only choose one dominant frequency to divide the whole time series data. But, due to the complexity of physical movement (i.e., each step is not exactly the same; some occasional movement can cause interference) it is not possible to use one frequency to extract each cycle without dislocation. However, in the long term, the set of cycles shows periodicity, as shown in Figure 7. In our opinion, that such a "greater period" exists among extracted cycles is similar to the concept of precession. As a result, we believe cycle series as a whole will contain all features needed. Therefore, our method of cycle extraction will not cause feature losses.

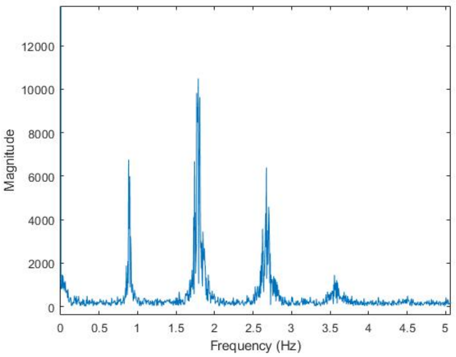


Fig 6a. Part of the Fourier transformation of Fig 5

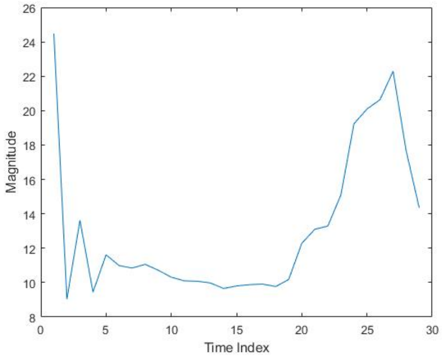


Fig 6b. Single motion cycle in time-domain

4.4 Cycle Data Visualization

We use two methods of data visualization in this paper, the Wigner-Ville distribution and Recurrence plot, to generate multiple graphs for the Convolutional Neural Network to perform identification.

4.4.1 Wigner–Ville Distribution

One way to analyze a multi-component signal like the motion cycle is through its instantaneous frequency (IF). The IF estimation method identifies the local extrema in the time-frequency space. In this step, the valid peaks are the local extrema which have values higher or lower than a predefined threshold. The idea behind this is that the IF of a component of a signal (where the energy of the

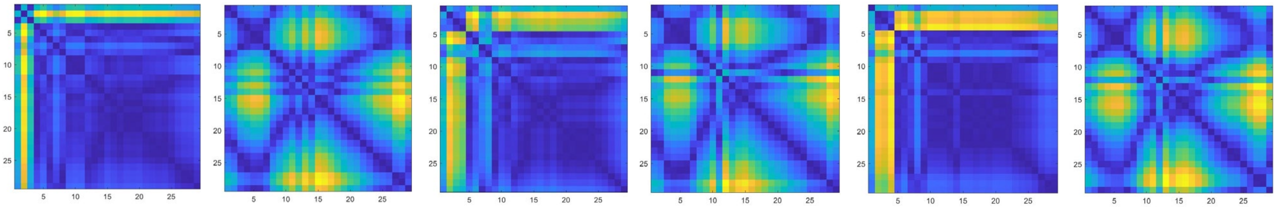


Fig 7. Six continuous, Recurrence plots of Subject 3; notice that there is a period of 2 among the plot series

signal is concentrated) is observable in the time-frequency space as a ridge which describes the IF. So, to use IF features for identification, we must first map our time-domain cycle signal into time-frequency space, which can be done through time frequency transformation performed by the time frequency distribution (TFD) method.

There exist several TFD methods, such as the short-time Fourier transform, wavelet transform, or Hilbert–Huang transform However, we chose the Wigner-Ville distribution (WVD). WVD has been widely used for time-frequency analysis with good performance. We chose it because its relative high clarity (resolution).

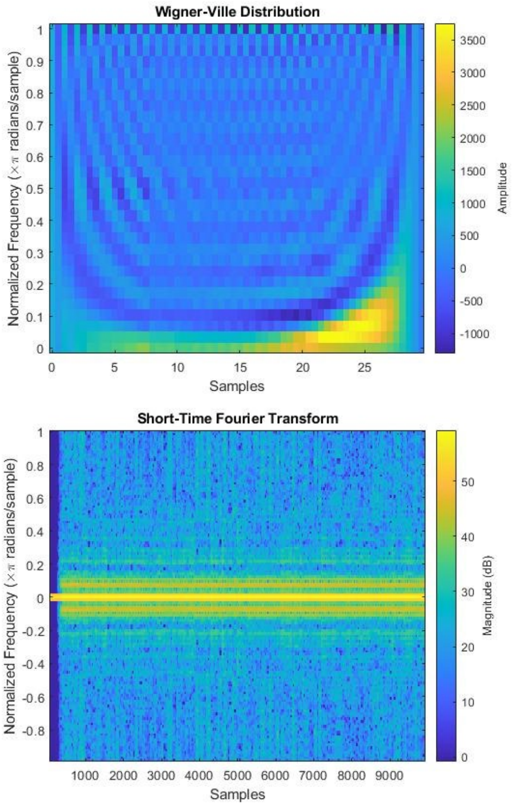


Fig 8. The upper is the WVD of a motional cycle signal. The lower is the STFT of a whole signal. Notice that, if under the same time scale, STFT will have a much lower resolution.

4.4.2 Recurrence plot

Recurrence plot is another common data visualization method for measuring the time constancy of dynamic systems [16]. Recurrence plot is a plot in phase space which shows the trajectory which visit similar spaces in different time moments. In other words, recurrence plots present the similarity between each moment in time and all other moments in the time series. Specifically, recurrence plots are a matrix generated by calculating the Euclidean distance between values of each pair of time moment *i* and *j*, where *i* and *j* are both in the time series. The term "recurrence" indicates that the plot shows the times when a state of the dynamic system represented by the time series data recurs.

Recurrence plot can be mathematically expressed as equation below:

$$R_{i,j} = \Theta\left(\varepsilon_i - \| \overrightarrow{x_i} - \overrightarrow{x_j} \|\right),\ \overrightarrow{x_i} \in \mathbb{R}_m,\ i,j = 1, \dots, N \qquad (4)$$

where N is the number of considered states, x_i, ε_i is a threshold distance, $\|\cdot\|$ is a norm, and $\Theta(\cdot)$ is the Heaviside function.

Due to the Heaviside function, typical Recurrence plot is under binarization by the threshold. However, we consider that such a threshold is hard to choose and the binarization process may cause some feature losses. To prevent such problems, we gave up the binarization step and collected the colored distance matrix generated by MATLAB instead. An example of a plot we used is shown in Figure 9.

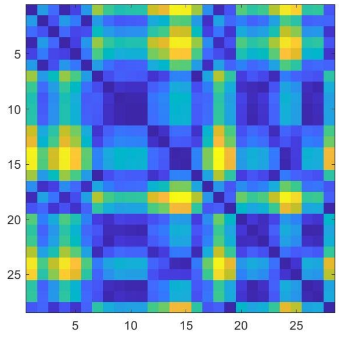


Fig 9. Recurrence plot (without binarization) of one motion cycle. The x-axis and y-axis are both in a time index which is the same length as the cycle.

4.5 Convolutional Neural Networks

When the time frequency representation of a subject had been extracted, we compared the results. We identified that patterns exist in the representations. We decide to use Convolutional Neural Networks (CNN) for feature extraction and identification.

CNN is a class of deep neural networks that is most commonly used in visual image analysis, pattern recognition, and natural language processing. CNN is designed to learn image features through backpropagation by using convolution layers, pooling layers, and fully connected layers.

Because we did not have enough data and resources to train a CNN model from scratch, we decided to use transfer learning for our project. Transfer learning (TL) is a machine learning method which reuses a pre-trained machine learning model on a new problem. It uses the "knowledge" that a model learned from a specific problem and applies it to a new problem which does not have much data with which to train a model. In general, transfer learning requires less training data and fewer computing resources during training in addition to being easier to use. We choose to use VGG16 as our base model.

VGG is a deep convolutional neural network proposed by K. Simonyan and A. Zisserman in their paper [11]. The model is known for its high accuracy in image classification. Its dataset contains over 14 million images that belong to 1000 classes. VGG contains 13 convolutions layers, 5 pooling layers, 3 fully connected layers, and 1 softmax output layer. All activation functions are the rectified Linear Units (ReLU) function. We altered the last few layers of the VGG16 model to achieve our goal.

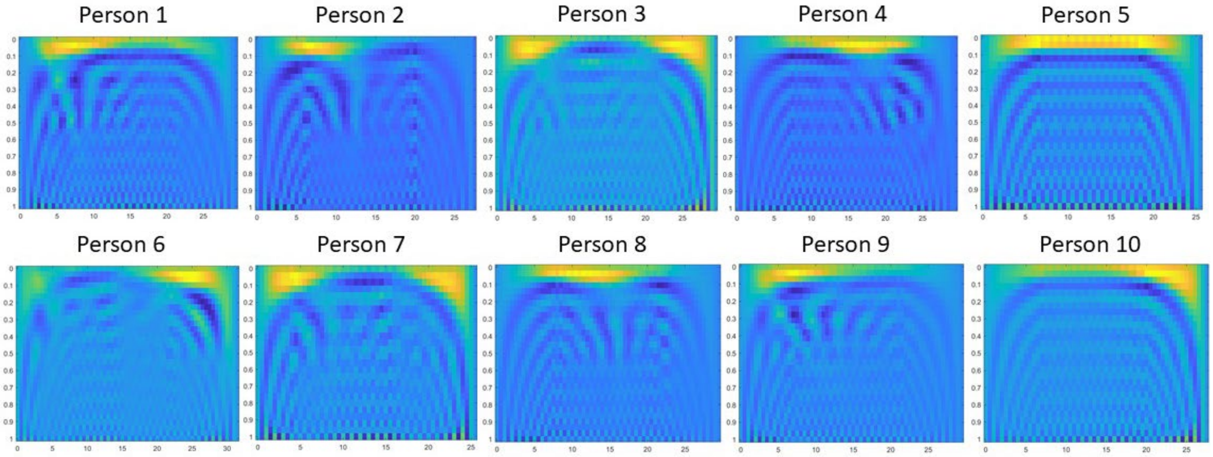


Fig 10. Wigner-Ville distribution (WVD) of one motion cycle from 10 different subjects

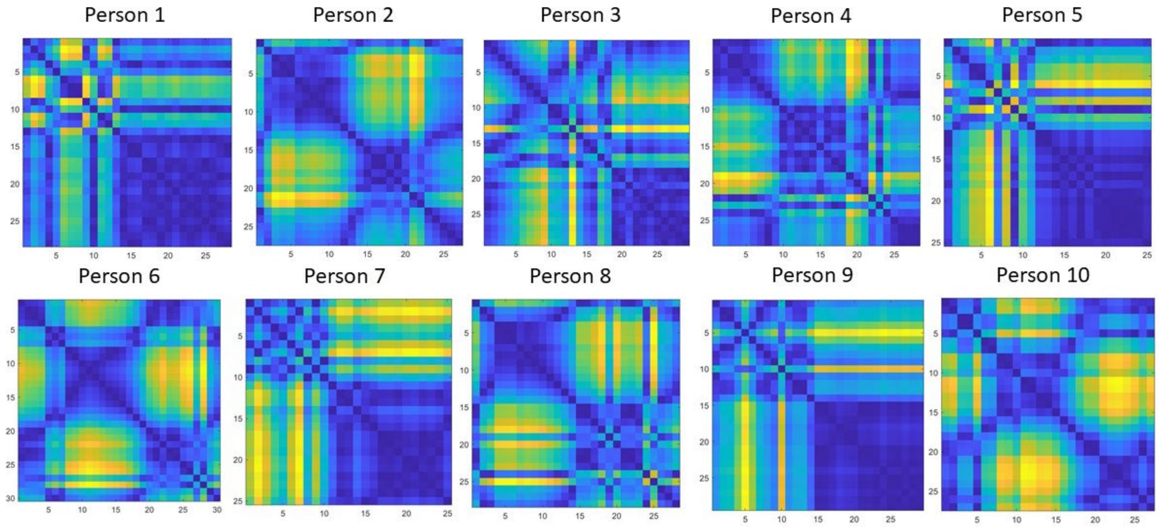
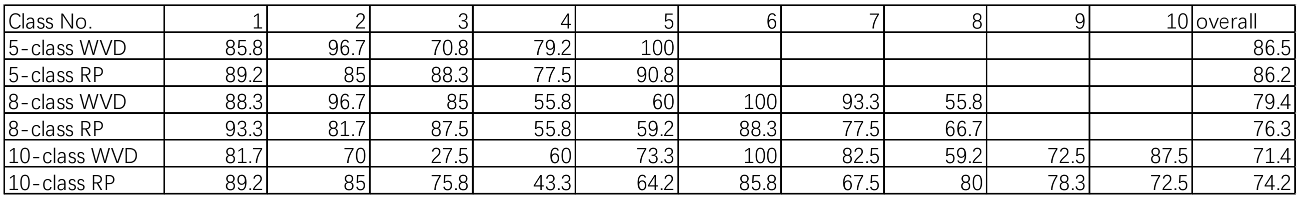


Fig 11. Recurrence plots (without binarization) of one motion cycle from 10 different subjects. For each picture, the x-axis and y-axis are both in the time index



Class No.	1	2	3	4	5	6	7	8	9	10	overall
5-class WVD	85.8	96.7	70.8	79.2	100						86.5
5-class RP	89.2	85	88.3	77.5	90.8						86.2
8-class WVD	88.3	96.7	85	55.8	60	100	93.3	55.8			79.4
8-class RP	93.3	81.7	87.5	55.8	59.2	88.3	77.5	66.7			76.3
10-class WVD	81.7	70	27.5	60	73.3	100	82.5	59.2	72.5	87.5	71.4
10-class RP	89.2	85	75.8	43.3	64.2	85.8	67.5	80	78.3	72.5	74.2

Table 1. The accuracy (%) of each identification based on confusion matrices

5 Experiment

We used the acceleration data from shin-mounted sensors on 10 people during walking. These data were stored in 10 .csv files. Each file contains over 30,000 data points. We picked the first 20,000 points for analysis. We used MATLAB to perform data pre-processing, cycle extraction, and TFD transforms. MATLAB has a built-in fast Fourier transform function (fft), which we used for cycle frequency detection. It also has a Wigner-Ville distribution (wvd) function which supports image generation in the signal processing toolbox. We used the wvd() function to generate time-frequency space images for motion cycles. The recurrence plot is generated by a loop in MATLAB. Then, we collected these images to build training, validation, and test sets for the fine-tuned VGG16 network. Due to the "blank" in the beginning of each of the 10 time-series signals, we started image generation after 20 times of the cycle length. Eventually, we generated 1200 images for each subject (600 images using WVD and the others using recurrence plot), with a total of 12000. The example of these images is shown in Figure 10 and Figure 11. For each of the 600 images using the same visualization method of each subject, we selected 360 images as a training set, 120 images as a validation set, and 120 images as a dataset.

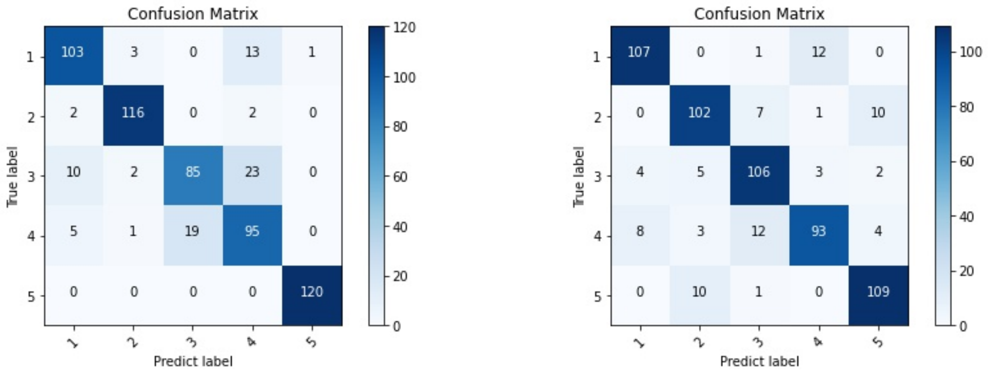
The VGG16 network we use is from the Keras application. Keras is a Python deep learning API running on TensorFlow, a machine learning platform. We fine-tuned the original VGG16 network for our experiment. The original VGG16 network has a fully-connected layer called "prediction layer" with a depth of 1000 at the end. This is used to train on ImageNet—a functionality we did not require. We changed these layer's depths to 5,8 and 10 according to the scenario since there were only so many classes in the dataset. Additionally, we added a dropout layer before the prediction layer with loss rate of 0.2. We used this dropout layer to help reduce overfitting.

We used the fine-tuned VGG16 network for 3 scenarios in our study: the 5-class, 8-class, and 10-class identification. In all scenarios, the network was trained for 20 epochs. The batch size for training and validation was 30. The batch size of testing is equal to the number of data classes according to the scenario. We chose 20 epochs because the training process shows that the gains become increasingly smaller, and the process curve becomes increasingly flat after 20 epochs. The training results of the last several epochs are shown below:

```
Epoch 16/20
120/120 [==============================] - 32s
268ms/step - loss: 0.8795 - accuracy: 0.6761 - val_loss: 0.7840
- val_accuracy: 0.7008
Epoch 17/20
120/120 [==============================] - 32s
266ms/step - loss: 0.8388 - accuracy: 0.6953 - val_loss: 0.6950
- val_accuracy: 0.7083
Epoch 18/20
120/120 [==============================] - 32s
267ms/step - loss: 0.8312 - accuracy: 0.6978 - val_loss: 0.8434
- val_accuracy: 0.6975
Epoch 19/20
120/120 [==============================] - 32s
266ms/step - loss: 0.8221 - accuracy: 0.6956 - val_loss: 0.6034
- val_accuracy: 0.7075
Epoch 20/20
120/120 [==============================] - 32s
266ms/step - loss: 0.7808 - accuracy: 0.7236 - val_loss: 1.0131
- val_accuracy: 0.7100
```

6 Results

The final results of the experiment were the testing results of the CNN network, expressed as confusion matrices. For each matrix, the Predict Label stands for the prediction made by our neural network and the True Label indicates the true identity of images in the test set. So, if the predictions were all correct for one image class, the number of images of that class in the test set should be shown on the diagonal of the confusion matrix. From the experiment, we got 6 confusion matrices: 2 matrices used WVD images and the others used Recurrence plots for each of the 5-class, 8-class, and 10-class identifications. The accuracies of each of the identifications are shown in Table 1.

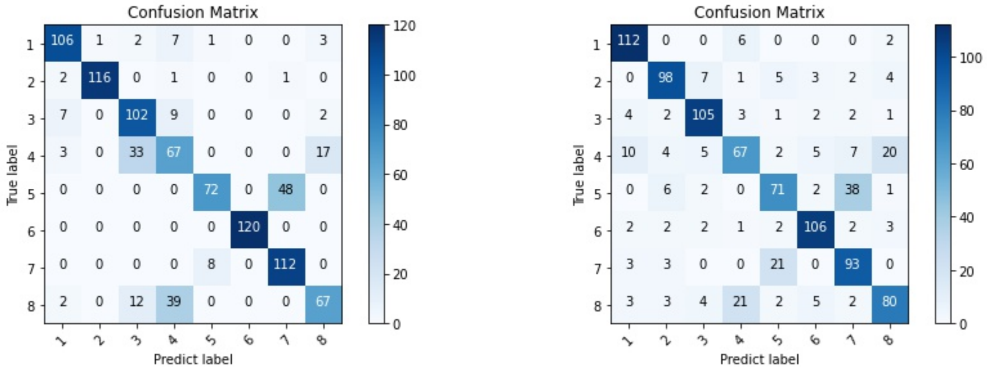


	1	2	3	4	5
1	103	3	0	13	1
2	2	116	0	2	0
3	10	2	85	23	0
4	5	1	19	95	0
5	0	0	0	0	120

	1	2	3	4	5
1	107	0	1	12	0
2	0	102	7	1	10
3	4	5	106	3	2
4	8	3	12	93	4
5	0	10	1	0	109

Fig 12. 5-class identifications

Left: based on Wigner-Ville distribution Right: based on Recurrence Plot

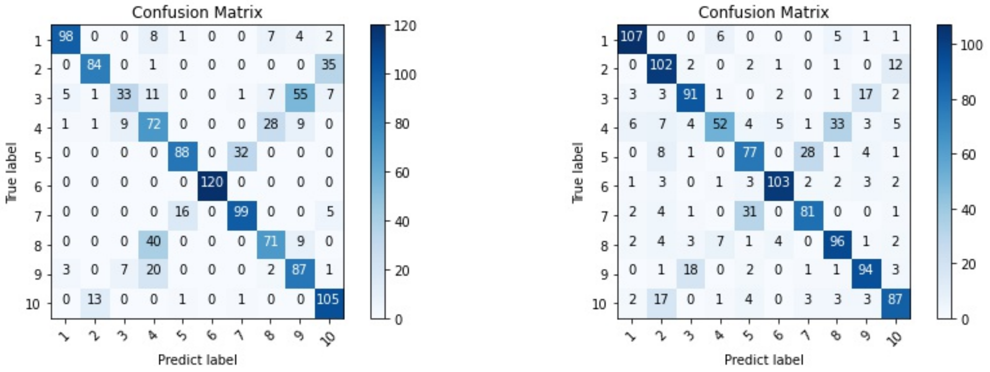


	1	2	3	4	5	6	7	8
1	106	1	2	7	1	0	0	3
2	2	116	0	1	0	0	1	0
3	7	0	102	9	0	0	0	2
4	3	0	33	67	0	0	0	17
5	0	0	0	0	72	0	48	0
6	0	0	0	0	0	120	0	0
7	0	0	0	0	8	0	112	0
8	2	0	12	39	0	0	0	67

	1	2	3	4	5	6	7	8
1	112	0	0	6	0	0	0	2
2	0	98	7	1	5	3	2	4
3	4	2	105	3	1	2	2	1
4	10	4	5	67	2	5	7	20
5	0	6	2	0	71	2	38	1
6	2	2	2	1	2	106	2	3
7	3	3	0	0	21	0	93	0
8	3	3	4	21	2	5	2	80

Fig 13. 8-class identifications

Left: based on Wigner-Ville distribution Right: based on Recurrence Plot



	1	2	3	4	5	6	7	8	9	10
1	98	0	0	8	1	0	0	7	4	2
2	0	84	0	1	0	0	0	0	0	35
3	5	1	33	11	0	0	1	7	55	7
4	1	1	9	72	0	0	0	28	9	0
5	0	0	0	0	88	0	32	0	0	0
6	0	0	0	0	0	120	0	0	0	0
7	0	0	0	0	16	0	99	0	0	5
8	0	0	0	40	0	0	0	71	9	0
9	3	0	7	20	0	0	0	2	87	1
10	0	13	0	0	1	0	1	0	0	105

	1	2	3	4	5	6	7	8	9	10
1	107	0	0	6	0	0	0	5	1	1
2	0	102	2	0	2	1	0	1	0	12
3	3	3	91	1	0	2	0	1	17	2
4	6	7	4	52	4	5	1	33	3	5
5	0	8	1	0	77	0	28	1	4	1
6	1	3	0	1	3	103	2	2	3	2
7	2	4	1	0	31	0	81	0	0	1
8	2	4	3	7	1	4	0	96	1	2
9	0	1	18	0	2	0	1	1	94	3
10	2	17	0	1	4	0	3	3	3	87

Fig 14. 10-class identifications

Left: based on Wigner-Ville distribution Right: based on Recurrence Plot

The results of **5-class identification** are shown in Figure 12 below. For the WVD method, the accuracy of each class was as follows: 85.8% for Class 1, 96.7% for Class 2, 70.8% for Class 3, 79.2% for Class 4, and 100% for Class 5. The overall accuracy was **86.5%.** For the Recurrence Plot method, the accuracy of each class was as follows: 89.2% for Class 1, 85% for Class 2, 88.3% for Class 3, 77.5% for Class 4, 90.8% for Class 5. The overall accuracy was **86.2%.**

The results of **8-class identification** are shown in Figure 13 below. For WVD method, the accuracy of each class was as follows: 88.3% for Class 1, 96.7% for Class 2, 85% for Class 3, 55.8% for Class 4, 60% for Class 5, 100% for Class 6, 93.3% for Class 7, and 55.8% for Class 8. The overall accuracy was **79.4%**. For Recurrence Plot method, the accuracy of each class was as follows: 93.3% for Class 1, 81.7% for Class 2, 87.5% for Class 3, 55.8% for Class 4, 59.2% for Class 5, 88.3% for Class 6, 77.5% for Class 7, and 76.3% for Class 8. The overall accuracy was **86.2%.**

The results of **10-class identification** are shown in Figure 14 below. For WVD method, the accuracy of each class was as follows: 81.7% for Class 1, 70% for Class 2, 27.5% for Class 3, 60% for Class 4, 73.3% for Class 5, 100% for Class 6, 82.5% for Class 7, 59.2% for Class 8, 72.5% for Class 9, and 87.5% for Class 10. The overall accuracy was **71.4%.** The accuracy of each class was as follows for the recurrent plot method: 89.2% for Class 1, 85% for Class 2, 75.8% for Class 3, 43.3% for Class 4, 64.2% for Class 5, 85.8% for Class 6, 67.5% for Class 7, 80% for Class 8, 78.3% for Class 9, and 72.5% for Class 10. The overall accuracy was **74.2%.**

The results showed that the current system's performance was not high enough for commercial use (i.e., above 95%), but was acceptable overall (above 70%). For each identification, the accuracies among each class were not consistent: The accuracy of each class varied from 50% to 100%. The situation became even more extreme in the 10-class WVD identification; the accuracy of Class 3 dropped to 27.5%, which constituted a failure. We think the inconsistencies may be due to the physical differences between the subjects. The 10 subjects we chose vary in gender, age, height, weight, and walking environment. Differences in these areas may cause more interference with certain identification tasks. For example, by checking the video record provided by the HAR dataset, we found that some subjects with older ages or walking in complex environments like mountain paths (i.e., Figure 15) were more likely to make occasional stops than young subjects walking in urban areas. Such occasional, non-periodic movements called "motion artifacts" in medical science were challenging for our system to identify.

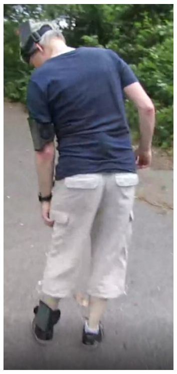


Fig 15. An elderly subject stopped to adjust his shoes during data collection

The results showed no major difference in performance between the WVD method and the recurrence plot method for all three identification scenarios. For the 5-class and 8-class identification, the WVD method has a slightly higher accuracy. However, for the 10-class identification, the accuracy of the recurrence plot method is slightly higher. However, there is a prediction with extremely low performance in the 10-class WVD identification; the accuracy of Class 3 dropped to 27.5% (shown in Figure 14 Left). As a result, we believe the recurrence plot method is more stable for identification with more classes.

The results show that the accuracy of our system decreases when dealing with more subjects. As Figure 16 shows, the overall accuracy drops from above 85% of the 5-class identification to below 75% of the 10-class identification. This might be because, as the number of subjects increases, the possibility for the system to encounter similar feature patterns increases. After all, the motion cycles of human activities always share some sort of common feature, especially when the physical condition of two subjects (e.g., height, weight) are similar. This problem of common features makes the application of motion feature identification methods in daily life more challenging.

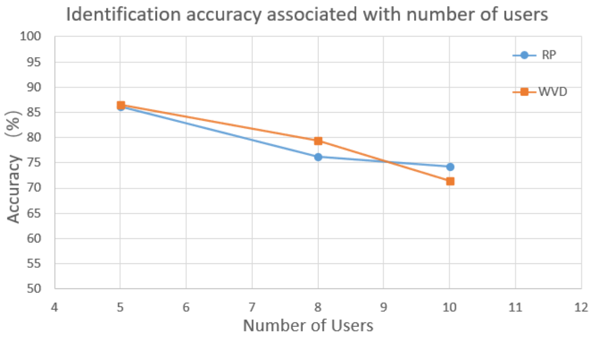
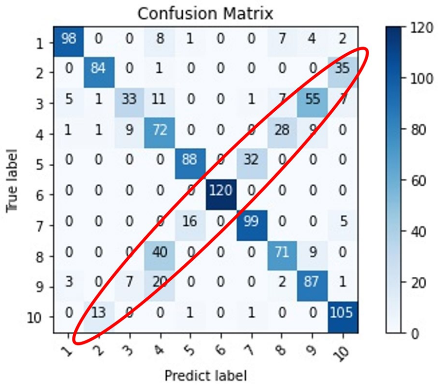


Fig 16. The overall accuracy decreases when the number of subject increases.

Additionally, we observed from the confusion matrices of the 8-class and 10-class identifications that the wrong predictions seem to be located in the matrices according to some rules: most of the incorrect predictions are located on a line which is perpendicular to the diagonal of the confusion matrix and intersects the diagonal at (6,6), where the true predictions of Class 6 should lie. The reason for this phenomenon is currently unknown. We only have a basic guess: it might be because of the inside structure and the weights of our CNN. Illustrating this phenomenon may require a deep comprehension of the workflow inside our VGG16 network.



Confusion Matrix

True label \ Predict label	1	2	3	4	5	6	7	8	9	10
1	98	0	0	8	1	0	0	7	4	2
2	0	84	0	1	0	0	0	0	0	35
3	5	1	33	11	0	0	1	7	55	7
4	1	1	9	72	0	0	0	28	9	0
5	0	0	0	0	88	0	32	0	0	0
6	0	0	0	0	0	120	0	0	0	0
7	0	0	0	0	16	0	99	0	0	5
8	0	0	0	40	0	0	0	71	9	0
9	3	0	7	20	0	0	0	2	87	1
10	0	13	0	0	1	0	1	0	0	105

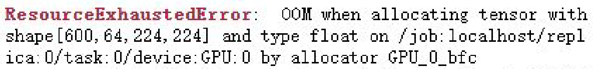
Fig 17. A 10-class identification example shows that most wrong predictions lie within the red circle

7 Future Works

According to our experience, results analysis, there are several points that we need to focus on in future work:

1. We need more computing power to deal with more data, creating a more realistic scenario.

   The average time we used to train the fine-tuned VGG16 network was 5-7 minutes for 5-class identification, 7-9 minutes for 8-class identification, and 10-13 minutes for 10-class identification. The computing time of our system was high, limiting its use in real-world scenarios with a greater number of subjects. However, when we tried to decrease the computing time by increasing the batch size, we received an Out Of Memory (OOM) error. The calculating power of our hardware was insufficient to decrease the computing time or process data from a greater number of subjects. We need GPUs with higher performance and larger memory in the future.



```
ResourceExhaustedError: OOM when allocating tensor with
shape[600,64,224,224] and type float on /job:localhost/repl
ica:0/task:0/device:GPU:0 by allocator GPU_0_bfc
```

Fig 16. We received an OOM error when we tried to increase batch size to shorten training time.

2. Motion feature identification may not be effective based solely on acceleration data from a single body part.

   In this paper, our work is based on walking data from a shin-mounted accelerometer. As we mentioned in section 4.1, these data are only a small part of the whole DataSet-RealWorld (HAR). To increase the accuracy of our system, we could combine data samples from different sensor locations or different sensors on the same person. Alternatively, we could use the same sensor's data but add more activities. These methods require us to adjust the CNN classifier to do the "sensor fusion" task.

3. We could improve our CNN based classifier.

   There are several ways we can improve the accuracy of our CNN classifier. First, we could make deeper modifications to the current VGG16 network. Also, we could design and train a CNN model with a larger dataset specifically for our MFI system.

   The VGG16 network we used for classification could still be further tuned to increase prediction accuracy. In this study, we only made basic modifications to the VGG16 network. Since our 2-D map images are unique, the pre-trained weight for CNN which is useful for common computer vision tasks may not be beneficial to us. We might need to deeply modify the structure of the neural network to make it more suitable for our mission after we learn more.

   Designing and training a neural network solely by our own data, which requires far more data than we currently have, may improve performance. The current HAR dataset we used in this paper is not large enough to train a CNN model from scratch. With a larger dataset, we can design and train a CNN model specifically for 2-D feature maps in our IMF system, which may result in higher accuracy.

4. Using methods like LIME to explain the workflow inside the CNN classifier and illustrate the phenomenon mentioned at the end of Section 6.

   We could use a method like LIME to show the workflow inside our CNN classifier instead of threating our classifier as a black box. LIME is a novel explanation technique that explains the predictions of any classifier in an interpretable and faithful manner by learning an interpretable model locally around the prediction [17]. With its help, we can know how our CNN makes the final prediction step by step. Thus, we can discover in which step the classifier makes the wrong decision and places most wrong predictions into the line in the confusion matrices mentioned in section 6.

5. We could further improve the resolution of 2-D feature map images.

   Even with our data preprocessing method, it is possible that some of the key motion features are lost during. We want to test other ways to transfer time series data into images that contain more key features to feed our CNN model. For example, these methods include smoothed-WVD or Gaussian Kernel.

6. We could perform data limitation analysis.

   Despite the uniqueness of an individual's motion feature data, related studies mentioned that a change in weight (by carrying weight in hand or on one's back) or wearing different shoes change MFI data [4]. We need a more controlled dataset to help us further analyze these kinds of factors that affect the identification process. An ideal dataset would still contain accelerator data, but all exercises would be performed in a controlled environment such as on a treadmill. There would be fewer variables using the treadmill since walking or running speed and exercise condition can be controlled. Additionally, there would be fewer motion artifacts. In this case, data from the same person with different shoes or weights can be collected easily; this might help us determine the hidden factors affecting our classification results.

In all these cases, we must do more research to further explore this area of study.

8 Conclusion

In this paper, we demonstrate a method of using motion cycle extraction and visualization to identify individuals using CNN based on motion sensor data in wearable devices. We divided the raw data series by the cycle length we extracted. Then, we used two different data-visualization methods to generate graphs for each cycle. We used these graphs to train the CNN classifier and make predictions.

The results of this study indicated the CNN transfer leaning predictions were acceptable, but not ready for commercial use.

Despite our method's shortcomings, we believe our work shows the principal feasibility of motion feature identification. Additionally, this work provides a solid foundation for further exploration of MFI technology. In future works, we shall analyze various avenues to improve the accuracy of the CNN's predictions.

ACKNOWLEDGMENTS

We would like to thank Dr. Ortiz for his help and support.

REFERENCES

[1] Ahmadi N, Akbarizadeh G (2018) Iris tissue recognition based on GLDM feature extraction and hybrid MLPNN-ICA classifier. Neural Comput Appl. doi:10.1007/s00521-018-3754-0

[2] Li J, Qiu T, Wen C, Xie K, Wen FQ. Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion. Sensors (Basel). 2018;18(7):2080. Published 2018 Jun 28. doi:10.3390/s18072080

[3] Zeng, F., Hu, S. & Xiao, K. Research on partial fingerprint recognition algorithm based on deep learning. Neural Comput & Applic 31, 4789–4798 (2019). doi: 10.1007/s00521-018-3609-8.

[4] D. Gafurov, E. Snekkenes and P. Bours, "Improved Gait Recognition Performance Using Cycle Matching," 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, Perth, WA, 2010, pp. 836-841, doi: 10.1109/WAINA.2010.145.

[5] Lee, T.K.M., Belkhatir, M. & Sanei, S. A comprehensive review of past and present vision-based techniques for gait recognition. Multimed Tools Appl 72, 2833–2869 (2014). doi:10.1007/s11042-013-1574-x

[6] Le M.J., Ingo S. The smartphone as a gait recognition device impact of selected parameters on gait recognition; Proceedings of the 2015 International Conference on Information Systems Security and Privacy (ICISSP); Angers, France. 9–11 February 2015.

[7] Sposaro F, Tyson G. iFall: an Android application for fall monitoring and response. Annu Int Conf IEEE Eng Med Biol Soc. 2009;2009:6119-22. doi: 10.1109/IEMBS.2009.5334912. PMID: 19965264.

[8] Yurtman A, Barshan B. Activity Recognition Invariant to Sensor Orientation with Wearable Motion Sensors. Sensors (Basel). 2017 Aug 9;17(8):1838. doi: 10.3390/s17081838. PMID: 28792481; PMCID: PMC5579846.

[9] L. Rong, D. Zhiguo, Z. Jianzhong and L. Ming, "Identification of Individual Walking Patterns Using Gait Acceleration," 2007 1st International Conference on Bioinformatics and Biomedical Engineering, Wuhan, 2007, pp. 543-546, doi: 10.1109/ICBBE.2007.142.

[10] R. Subramanian et al., "Orientation invariant gait matching algorithm based on the Kabsch alignment," IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015), Hong Kong, 2015, pp. 1-8, doi: 10.1109/ISBA.2015.7126347.

[11] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[12] J. R. Kwapisz, G. M. Weiss and S. A. Moore, "Cell phone-based biometric identification," 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington, DC, 2010, pp. 1-7, doi: 10.1109/BTAS.2010.5634532.

[13] N. Abid, P. Kozlow and S. Yanushkevich, "Detection of Asymmetric Abnormalities in Gait using Depth Data and Dynamic Bayesian Networks," 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 2018, pp. 762-767, doi: 10.1109/ICSP.2018.8652291.

[14] Dostál, O., Procházka, A., Vyšata, O. et al. Recognition of motion patterns using accelerometers for ataxic gait assessment. Neural Comput & Applic (2020).

[15] H. M. Thang, V. Q. Viet, N. Dinh Thuc and D. Choi, "Gait identification using accelerometer on mobile phone," 2012 International Conference on Control, Automation and Information Sciences (ICCAIS), Ho Chi Minh City, 2012, pp. 344-348, doi: 10.1109/ICCAIS.2012.6466615.

[16] J.-P Eckmann, , S. Oliffson Kamphorst, and D Ruelle. "Recurrence Plots of Dynamical Systems".Europhysics Letters (EPL) 4, no.9 (1987): 973–977.

[17] Tulio Ribeiro, M., Singh, S., and Guestrin, C. 2016. ``Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv e-prints, p.arXiv:1602.04938.

[18] Human Activity Recognition DataSet-RealWorld (HAR) 2016, https://sensor.informatik.uni-mannheim.de/index.html#dataset_realworld