Mathematical Biosciences and Engineering

MBE, 20(5): 8049–8067.
DOI: 10.3934/mbe.2023349
Received: 15 January 2023
Revised: 07 February 2023
Accepted: 13 February 2023
Published: 23 February 2023

http://www.aimspress.com/journal/MBE

Research article

A double-channel multiscale depthwise separable convolutional neural network for abnormal gait recognition

Xiaoguang Liu[1, 2], Yubo Wu[1, 2], Meng Chen[1, 2], Tie Liang[1, 2], Fei Han[1, 2] and Xiuling Liu[1, 2,]*

[1] College of Electronic and Information Engineering, Hebei University, Baoding, China
[2] Key Laboratory of Digital Medical Engineering of Hebei Province, Hebei University, Baoding, China

* **Correspondence:** Email: liuxiuling121@hotmail.com.

Abstract: Abnormal gait recognition is important for detecting body part weakness and diagnosing diseases. The abnormal gait hides a considerable amount of information. In order to extract the fine, spatial feature information in the abnormal gait and reduce the computational cost arising from excessive network parameters, this paper proposes a double-channel multiscale depthwise separable convolutional neural network (DCMSDSCNN) for abnormal gait recognition. The method designs a multiscale depthwise feature extraction block (MDB), uses depthwise separable convolution (DSC) instead of standard convolution in the module and introduces the Bottleneck (BK) structure to optimize the MDB. The module achieves the extraction of effective features of abnormal gaits at different scales, and reduces the computational cost of the network. Experimental results show that the gait recognition accuracy is up to 99.60%, while the memory size of the model is reduced 4.21 times than before optimization.

Keywords: abnormal gait recognition; convolutional neural network (CNN); MDB, BK structure; double-channel network

1. Introduction

Gait is an important biomedical indicator, since abnormal or irregular gait can both predict and indicate health problems, and the classification of abnormal gait is important for detecting weak body parts and diagnosing diseases. Gait impairment is one of the early predictive indicators of fall risk and physical disability in the elderly [1,2]. Among children with cerebral palsy, gait is monitored to guide and improve the feasibility of adaptive walking strategies during the rehabilitation training of children [3]. In addition, irregularities in an individual's walking pattern have been shown to be accurate diagnostic indicators for a range of neurological disorders, such as Parkinson's disease [4] and Huntington's disease [5]. Therefore, monitoring gait can provide critical information for the early detection and diagnosis of diseases, as well as for identifying the risk of injury for vulnerable groups, such as the elderly.

Early gait recognition research relies on camera motion capture systems and floor pressure sensor technologies [6–9]. Due to the advent of Micro-Electro-Mechanical System technology, lower-cost and more applicable inertial sensors [10] are becoming increasingly mature for application in wearable portable devices, and are evolving as essential access to human gait information. With the development of artificial intelligence, more and more machine-learning-based gait recognition methods have been proposed, such as nearest neighbor classifiers [11,12], support vector machines (SVM) [13] and various clustering methods (K-Means, Hierarchical and more) [14]. However, the recognition accuracy of these models is significantly impacted by the manual extraction of gait features. Also, there is no reliable criterion for manually extracted gait features. Therefore, these approaches are both time-consuming and challenging. Huitzil et al. [15] proposed a fuzzy ontology for a gait recognition method that automatically computes gait features in the collected gait data sequences, thus making the dataset more reusable. This method gives us new inspiration.

In recent years, researchers have begun to extend deep learning methods to human gait recognition. In 2016, Gadaleta et al. [16] provided a framework for data acquisition and signal processing using CNN as feature extractors for authenticating users based on their gait features. This experiment showed that CNN could automatically extract gait features and obtain favorable recognition performance. In 2019, a study of 25 healthy subjects using inertial sensors to simulate hemiplegic, tiptoe and cross-threshold gait showed that the Long short term memory-Convolutional neural network [17] model extracted temporal and spatial information from gait data, respectively, to classify abnormal gait with an accuracy of up to 93.1%. In 2020, Chakraborty and Nandy [18] used two inertial sensors to collect walking gait data from healthy and cerebral palsy children. The method trained a multichannel one-dimensional convolutional neural network (1DCNN) model with the decomposed signal formed by discrete wavelet transform, which achieved 96.4% classification accuracy. In 2021, Jun et al. [19] proposed an abnormal gait classification method based on a multimodal mixed model with RNN and CNN. The method fed the collected sequential skeleton and foot pressure data into a multimodal mixed model to classify them, with a final model accuracy of 97.6%. Although these methods can achieve gait recognition well, they only extract gait features at a single scale, which may ignore some important fine spatial information, resulting in insufficient feature extraction, thus affecting the accuracy of gait recognition. Therefore, we propose the method of extracting multiscale features with MDB to solve the above problem. This method has a certain promotion effect on the development of gait feature extraction.

Numerous studies have shown that the depth [20] and width [21] of neural networks significantly affect the extraction of signal features. This paper proposes a double-channel multiscale depthwise convolutional neural network (DCMSDSCNN) to achieve feature extraction at different scales by designing MDB. It effectively reduces the model memory size and decreases the computational cost. In order to improve the performance of gait feature extraction, the network introduces a module with channel attention and spatial attention mechanisms [22,23] in tandem. An improved residual block (RB) [24,25] is used to realize that input data can be propagated forward across layers, weakening the strong connection between layers, and effectively improving the network prediction accuracy.

The main contributions of this paper are as follows:

1). This paper proposes a DCMSDSCNN network for extracting and fusing multiscale features of left and right legs, achieving more spatial feature information fusion in order to realize automatic gait recognition with better performance.

2). This study proposes the MDB, which replaces the standard convolution of convolution kernel size larger than 1 with DSC. Optimizing MDB using BK structures that change the number of dimensions to reduce the number of parameters and the amount of computation. The module ensures that the gait features of different scales are obtained from the temporal data, enriching the details of the feature map (FM) and improving the accuracy, and can reduce the model parameters.

3). Use improved RB to solve the information loss problem in training and reduce loss and computational consumption. Improve the saliency of feature information using an attention mechanism that uses channel attention in tandem with spatial attention to extract the critical feature information as much as possible.

4). In this paper, we evaluate the MSDSCNN network using time series data as input. The network contains two RBs with improved structure, which can effectively extract gait features, reduce losses and improve the accuracy of gait recognition.

The rest of the paper is organized as follows: Part II describes the experiments and data collection, the data processing methods and the dataset. Part III describes the model structure of deep learning and the efforts made to improve the accuracy of the model. The fourth part presents and analyzes the experimental results of abnormal gait classification. The conclusion and the direction of future work will be mentioned in the fifth part.

2. Materials and methods

2.1. Data acquisition system

This experiment used a miniature wireless inertial magnetic motion tracker (MTw) with a built-in 3D accelerometer and 3D gyroscope to collect gait data [26]. The triaxial accelerometer obtained the axial acceleration by measuring the force of the wearable device in one of the axes (X, Y or Z). At the same time, the acceleration measured by the accelerometer can reflect the motion of the wearable device user. A triaxial gyroscope worked by measuring the angle between the vertical axis of the gyroscope rotor and the device in a three-dimensional coordinate system to calculate the angular velocity. Therefore, the gyroscope can capture the angular velocity by measuring its own rotational state, which can also determine the movement state of users.

In consideration of recognition without disturbing normal activities, wearable sensors should be designed to be lighter, smaller and easier to wear. The Xsens sensor selected for this experiment measures $47\,mm \times 30\,mm \times 13\,mm$ and weighs 16 g, as shown in Figure 1. The MTw connects to the Awinda Station, and then wirelessly transmits real-time acceleration and angular velocity data to the PC. In the data acquisition experiments, the sensor collected data at a sampling frequency of 100 Hz. The computer environment for collecting the experimental data is shown in Table 1.

Table 1. Computer environment parameters.

Parameters	Configuration Information
CPU	Intel(R)Core(TM)i5-1135G7@2.40GHz
RAM	16.0 GB
Operating System	Microsoft Windows 10 (64 bit)

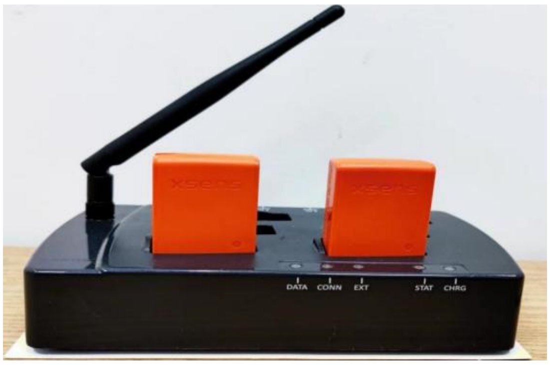


Figure 1. Wearable Xsens sensor.

2.2. Data acquisition experiments

In this experiment, five types of gait data were collected: normal gait, fast gait, slow gait, hemiplegic gait and analgesic gait. Twenty volunteers were recruited to participate in this experiment, at an average age of 25 years, all of whom were healthy subjects and could walk normally. We collected the data by requiring each subject to walk along a straight line on an indoor road of fixed length of 10 meters for one minute at a time, and each person needed to record five times. Before starting the experiment, volunteers were informed of the experimental requirements, and voluntarily completed a written informed consent form. The wearing position of the wearable Xsens sensor is shown in Figure 2.

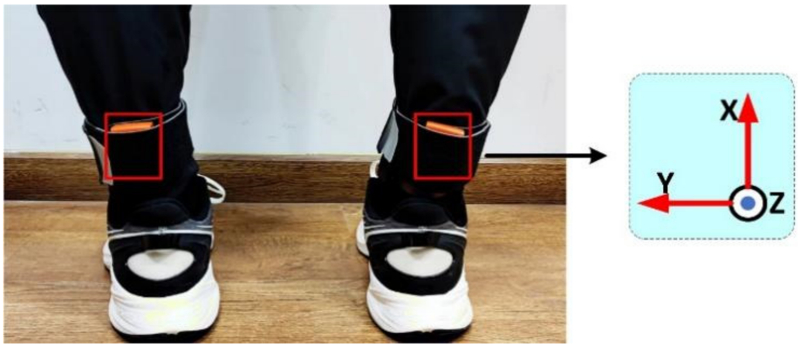
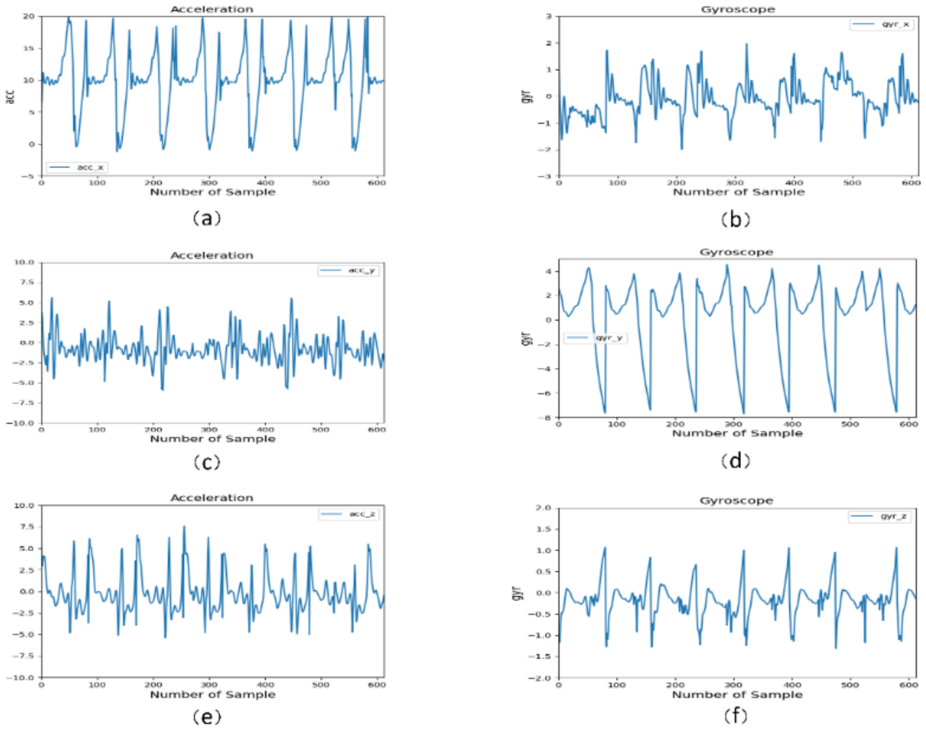


Figure 2. Wearable Xsens sensor installation.

The normal walking pattern required volunteers to walk at their most comfortable speed for the test, and the fast or slow walking patterns required volunteers to collect data faster or slower than the normal walking pattern.

Regarding abnormal gait, this experiment was performed to verify the abnormality of the right leg only, unilaterally (simulating one leg abnormality). The hemiplegic gait was characterized by stiff knees and legs, slight internal rotation and drooping of the feet, and bent toes. In the beginning, the instructions were to turn to the healthy side, lift the affected pelvis to lift the affected limb, and then use the affected hip as an axis, rub the straight legs and draw half a circle to the outside. Antalgic gait was usually an abnormal gait designed to avoid pain during walking, where the subject spent most of the standing time with the weight on the normal leg. The swing phase on the affected side was increased, and the stance (i.e., foot on the ground) phase was shortened to enable the normal leg to return to the ground quickly. During the experiment, analgesic gait required volunteers to remove their left shoe, and use the left side as the normal side to simulate the right analgesic gait. Figure 3 shows an example of the inertial signal collected during walking.



(a) (b) (c) (d) (e) (f)

Figure 3. Examples of inertial signals for gait acquisition: (a) acceleration on the x-axis, (b) angular velocity on the x-axis, (c) acceleration on the y-axis, (d) angular velocity on the y-axis, (e) acceleration on the z-axis and (f) angular velocity on the z-axis.

2.3. Data preprocessing

In this experiment, acceleration and gyroscope data were collected using two inertial sensors with a sampling frequency of 100 Hz. Each acceleration signal was recorded as a 3-dimensional vector $a = [a^X\ a^Y\ a^Z]$, where a^X, a^Y and a^Z are the accelerations acting along the X, Y and Z axes, correspondingly. Similarly, each gyroscope signal, g, was a 3-dimensional vector $g = [g^X\ g^Y\ g^Z]$, where g^X, g^Y, and g^Z represent the rotation rates around the X, Y and Z axes, respectively.

The second-order Butterworth filter was used for the filtering process because the collected data were disturbed by noise, and the Butterworth filter had the advantage of filtering uncertainty. The acceleration and gyroscope sequences were then combined into a 6-channel data stream. For the purpose of increasing the amount of training data, a sliding window with a constant length of 128 and 80% overlap is chosen to segment the data in this experiment, based on the experience of [27]. The dataset collected 23910 samples using the method described above. The amount of data for each of the five gait types is shown in Table 2. The window segmentation method could be defined as follows: suppose there existed a sample matrix S, the window size was α and the step size of the window was l (the step size was the window size minus the product of the constant length and the overlap rate), and the window segmentation algorithm was the following formula.

$$S_1 = \begin{bmatrix} a_1(t_i) & a_2(t_i) & \cdots & a_M(t_i) \\ a_1(t_{i+1}) & a_2(t_{i+1}) & \cdots & a_M(t_{i+1}) \\ \cdots & \cdots & \cdots & \cdots \\ a_1(t_{i+\alpha}) & a_2(t_{i+\alpha}) & \cdots & a_M(t_{i+\alpha}) \end{bmatrix} \tag{1}$$

$$S_2 = \begin{bmatrix} a_1(t_{i+l}) & a_2(t_{i+l}) & \cdots & a_M(t_{i+l}) \\ a_1(t_{i+l+1}) & a_2(t_{i+l+1}) & \cdots & a_M(t_{i+l+1}) \\ \cdots & \cdots & \cdots & \cdots \\ a_1(t_{i+l+\alpha}) & a_2(t_{i+l+\alpha}) & \cdots & a_M(t_{i+l+\alpha}) \end{bmatrix} \tag{2}$$

Table 2. The amount of data for each of the 5 gait types.

Gait type	Data amount
Normal gait	4489
Fast gait	4222
Slow gait	5589
Hemiplegic gait	5199
Analgesic gait	4411

Ultimately, we used a random shuffle approach, using 30% of the data for testing and 70% for training.

2.4. Model architecture

This paper proposes a DCMSDSCNN for abnormal gait recognition. Figure 4 depicts the architecture of the whole network model. The network model passes through two channels containing an RB. RB#1 and RB#2 are identical, where two MDBs are connected in series, and the MDBs are used to extract the multiscale features of abnormal gaits and reduce the number of parameters. The output features of the two channels are fused using the Concat function, and after the max pooling layer (MP) to compress the features, they enter RB#3, having three different MDBs in series. This section details the design of the network model and the work that has been done to improve the performance of the model.

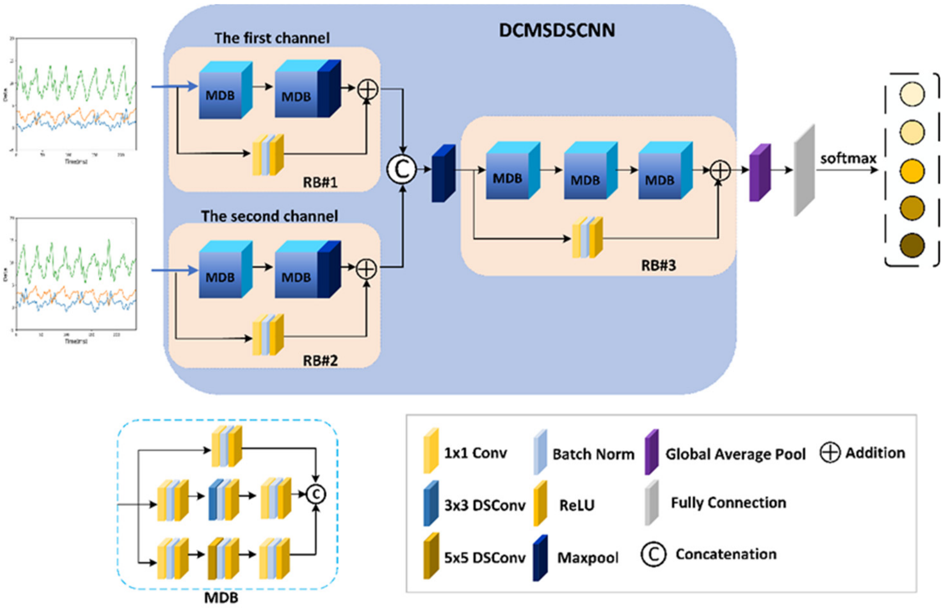


Figure 4. Architecture of DCMSDSCNN.

2.4.1. MDB

This paper proposes an MDB, which aims to extract multiscale features, obtain more fine information and capture better spatial features at the same time, but reduces the network parameters and reduces the computational cost. The MDB uses convolutional kernels of size 5×5, 3×3 and 1×1 for feature extraction of the time series data. The size of the first two convolution kernels, to extract the characteristics of the different sizes of the receptive field, uses 1-dimensional convolution for dimension reduction. Through MDB, richer gait information can be obtained. However, using standard convolution generates a large number of network parameters. Therefore, the network introduces DSC [28], which is increasingly receiving attention in the design of modern CNN architectures.

DSC can significantly reduce the number of network parameters and be used with almost no loss of accuracy or even an increase in accuracy. DSC divides standard convolution into two parts: depthwise convolution and pointwise convolution. The former convolves each channel of the input FM separately, thus capturing the spatial features of each channel. The latter integrates all the extracted spatial features and learns the channel correlation information of the input FM. Suppose the input image size is $H \times W \times C$, where H, W and C are the height, width and the number of channels of the input image, respectively. A convolution kernel of size $N \times N$ is used to perform depthwise convolution for each channel, and the parameters of depthwise convolution are calculated as follows:

$$K_{depthwise} = H \times W \times C \times N \times N. \tag{3}$$

For pointwise convolution, the output channel of the FM generated by depthwise convolution is expanded by an M convolution kernel of size 1×1. The cost of computing the parameters of the pointwise convolution is:

$$K_{pointwise} = H \times W \times C \times 1 \times 1 \times M. \tag{4}$$

Therefore, the computational volume of the DSC is the weighted value of the depthwise convolution and the pointwise convolution:

$$K_{separable} = K_{depthwise} + K_{pointwise} = H \times W \times C \times (N \times N + M). \tag{5}$$

For the standard convolution, the parameters are calculated as follows:

$$K_{standard} = H \times W \times C \times N \times N \times M. \tag{6}$$

From formulas 5 and 6, it can be known that, when N is greater than 1, the number of parameters of DSC is $(N^2 + M)/(N^2M)$ times smaller than the number of parameters of standard convolution. When N is equal to 1, the number of parameters of DSC is more than the standard convolution, so this method only replaces the standard convolution whose convolution kernel size is greater than 1. Figure 5 illustrates the fundamental structure of MDB.

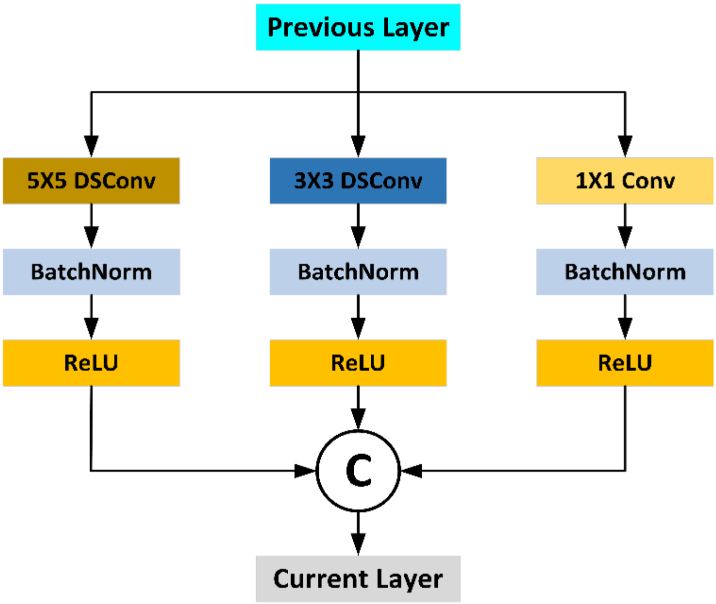


Figure 5. MDB infrastructure diagram.

Feature information at diverse scales can be extracted by MDB. However, adding convolutional kernels of multiple sizes increases the number of model parameters significantly, and the computational cost increases simultaneously. In order to address the above issue, the BK structure is cited in this paper to optimize the MDB. A basic example of the BK structure is shown in Figure 6.

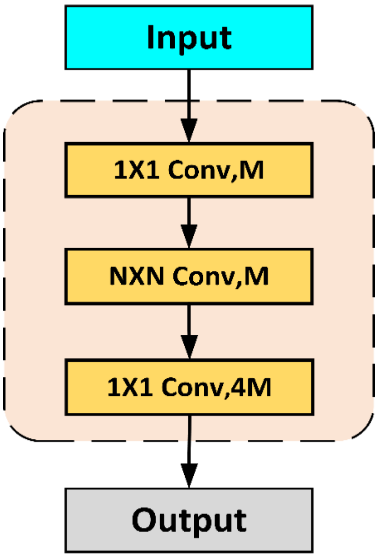


Figure 6. Example of BK structure base ($N \times N$ is the convolutional kernel size, where 4 M is the number of convolutional kernels).

The BK structure uses 1×1 convolution first to reduce, and then raise, the dimension, significantly reducing the computational consumption of one structural unit and improving the expressiveness of the calculated unit quantity of the network structure. At the same time, the BK structure introduces ReLU several times, dramatically improving the network structure's nonlinearity. This increase-and-decrease dimensionality achieves feature integration and model simplification across channels, further improving the overall performance of the neural network. Suppose the number of input samples is 128, the convolutional kernel size is 3×3 and the number of convolutional kernels is 128, the parameter amount of standard DSC is calculated as follows:

$$K_{separable} = X \times (N \times N + 4M) = 128 \times (3 \times 3 + 128) = 17536. \tag{7}$$

The number of depthwise separable convolutional parameters for applying the BK structure is:

$$K_{BKseparable} = X \times 1 \times 1 \times M + M \times (N \times N + M) + M \times 1 \times 1 \times 4M = 128 \times 32 + 32 \times (3 \times 3 + 32) + 32 \times 128 = 9504 \tag{8}$$

According to formulas 7 and 8, in the above assumptions, DSC with the BK structure applied has 46% fewer parameters than the standard DSC. This proves that using the BK structure can reduce the number of network parameters and save computational costs. The blue dashed box in Figure 4 shows the MDB with the BK structure applied, and the specific details are shown in Figure 7.

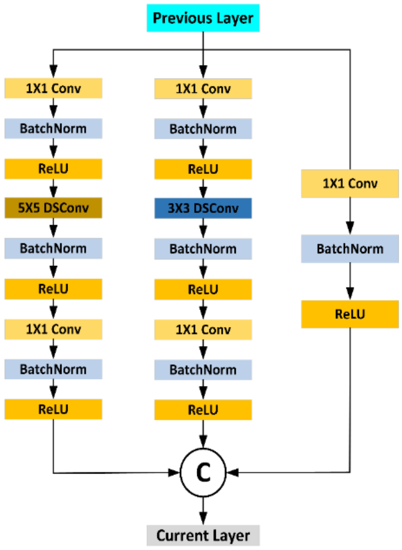


Figure 7. MDB diagram of the applied BK.

2.4.2. Convolutional Block Attention Module

Figure 8 shows the flow chart of the attention mechanism. In this article, the Convolutional Block Attention Module (CBAM) [22,23,29] is used to focus on the information of important features in the gait time series data through channel attention and spatial attention mechanisms, in series. The model applies CBAM on the convolutional output of each multiscale feature extraction block of both channels. Firstly, the input X_1 of the channel attention is subjected to a global average pooling (GAP) and a global max pooling (GMP) to reduce the size of the FM to 1×1, respectively, to focus on the channels of the FM. Next, the number of channels of the FM is compressed using a convolutional layer. Then, the number of channels is restored to the original number using a convolutional layer, and the two obtained FMs are summed. After that, each channel of the FM is weighted by the Sigmoid function, and the weight coefficients are multiplied with X_1 to obtain the final output X_2.

In the following step, X_2 is used as the input to the spatial attention mechanism. Two FMs are obtained by applying GMP and GAP to X_2 along the channel axis, respectively, and then they are stitched together using a Concat layer. The FMs are output through a convolution layer, a sigmoid function generates the spatial weight coefficients and the weight coefficients are multiplied with X_2 to, finally, obtain FMs with channel and spatial weights.

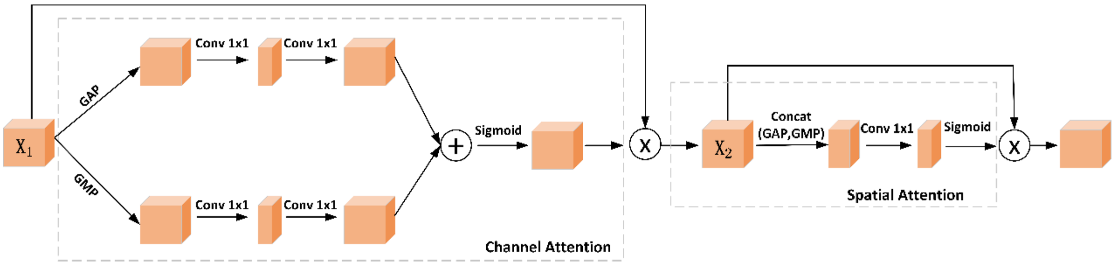


Figure 8. Flow chart of attention mechanism.

2.4.3. Multiscale depthwise separable convolutional neural network

The multiscale depthwise separable convolutional neural network (MSDSCNN) contains five different MDBs with 64, 96, 128, 160 and 192 convolution kernels. Each MDB uses 1×1, 3×3 and 5×5 convolutions to extract multiscale gait features. Each convolution layer uses Relu as the activation function to speed up convergence, prevent overfitting and adapts batch normalization before activation to prevent gradient disappearance. A jump connection is used at the ends of the first two and last three MDBs to form two RBs, RB#1 and RB#2 (equivalent to RB#1 and RB#3 in Figure 4). CBAM is applied to the convolutional output of each MDB of RB#1 for enhanced extraction of key gait information, which is not shown in Figure 4 for convenience. A max pooling layer of size 2×2 and step size 2 is added after the second MDB, removing redundant information and simplifying the network complexity. A GAP layer is attached after RB#2 to reduce many parameters, and the output results enter a fully connected layer (FL) with a 0.5 dropout rate to prevent overfitting. Finally, a softmax classifier is used for recognition. Figure 9 shows the network flow chart of MSDSCNN.

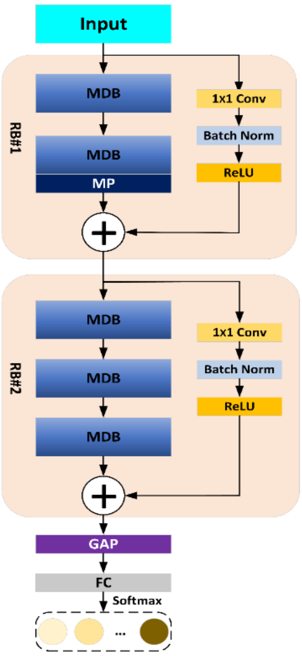


Figure 9. MSDSCNN network flow chart.

3. Results

3.1. Evaluation metrics

In this paper, evaluation metrics are introduced to validate the classification performance of the proposed DCMSDSCNN network. For gait recognition using deep learning, classifier performance can be measured by calculating accuracy, precision, recall and F1 score. Accuracy is the ratio of the number of samples correctly classified by the classifier to the total number of samples in a given test dataset. Higher accuracy indicates that the model has better predictive power. F1-score is used to balance precision and recall, and higher values represent better classifier performance. The above evaluation indexes are shown in formulas 9–12, where true positives, true negatives, false positives and false negatives are expressed as TP, TN, FP and FN, respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

$$F1\ score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{12}$$

3.2. Training method

In this work, we conduct experiments based on the open-source Python library TensorFlow. Since using stochastic gradient descent (SGD) often leads to a local minimum of the error, Adam [30] is used instead, combining the optimal performance of the AdaGrad and RMSProp algorithms, and providing optimization methods to solve sparse gradient and noise problems. The exponential decay rate of the moving average gradient (Beta_1) is 0.9, and the exponential decay rate of the moving average of the squared gradient (Beta_2) is 0.999. The learning step size of each iteration parameter has a certain range, which will not be too large because of the large gradient, and the parameter values are relatively stable.

3.3. Selection of hyperparameters

The impact of hyperparameters on the model is self-evident. The learning rate directly affects the ability of the model to converge and find the optimal solution. If the learning rate is set low, the update of the model parameters will be slow and require a long training time. However, if the learning rate is too large, it will lead to the direct oscillation of the loss function, which will not converge, and the classification ability of the model will be weakened. Too many iterations will not only take too long to train, but will also result in overfitting. The batch size is the number of samples sent to the model each time the network is trained. If the value chosen is too small, the statistics will not be representative, and the noise will increase, making it difficult for the network to converge. Nevertheless, if the value chosen is too large, the gradient direction will be stable and fall into the local optimal solution, affecting the model accuracy. Therefore, after several experimental verifications, the values of the hyperparameters applicable to the model in this paper are finally selected, as shown in Table 3.

Table 3. Values of hyperparameters.

Hyperparameters	Value Range	Value
Learning rate	0.001–0.005	0.002
Batch size	64–128	96
epoch	60–100	80
Dropout	0.2–0.5	0.5

3.4. Comparison of RB performance

Residual learning [24] was proposed to address the degradation phenomenon of deep networks, protect information integrity, reduce parameters and simplify the learning difficulty. In addition, the residual structure has a very efficient role in solving the problems of shallow architectures. This is so that the batch normalization layer can fully control the input information entering the activation layer to produce better accuracy. This paper proposes two improved RBs based on the original RBs, where one is the RB where the batch normalization layer is located after the Addition, and the other is the RB where the ReLU is located before the Addition. The Addition refers to the unit add operation, i.e., $X = F(x) + x$. Figure 10 shows the structure of the RB. Comparative experiments were done using the MSDSCNN model in the same configuration environment. The experimental results show that the ReLU before Addition structure is better for the recognition accuracy of the network. The comparison results are shown in Table 4.

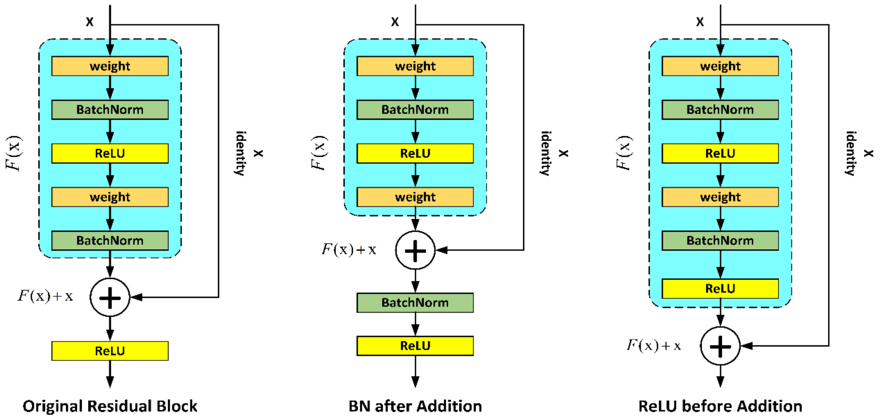


Figure 10. RB.

Table 4. RB performance comparison.

RB	Accuracy	Loss
Original RB	97.74%	0.1326
BN after Addition	98.02%	0.1035
ReLU before Addition	98.37%	0.0824

3.5. Classification results

3.5.1. Performance comparison of time series data models

In this experiment, the performance of the MSDSCNN proposed in this paper is evaluated using time series data (i.e., the gait data collected in 2.2). For verifying the effectiveness of the proposed method, this experiment compared the performance of gait recognition models using six methods [31–33]. To ensure the consistency of the experiments, the optimizers were all changed to Adam, the learning rate was set to 0.002, and the epoch number was 80. Table 5 shows the experimental results using different recognition methods.

The experimental results show that the existing advanced methods have good recognition accuracy on the dataset of the right leg collected in this study, which indicates that the abnormal gait data collected based on the acceleration and gyroscope of the wearable device can be distinguished. Although the MSDSCNN model applying the BK structure does not have the slightest memory size, it obtains higher recognition accuracy and F1-Score than the existing methods. The MSDSCNN model with the BK structure has 4.13 times less memory and 1.26% higher classification accuracy compared to the MSCNN model. Compared to the LSTM+CNN model, the classification accuracy is improved by 2.06%. This is due to the ability of the BK structure to reduce the number of parameters and computation by changing the dimensionality, and the MDB module can reduce the number of parameters while extracting the multiscale features, leading to improved accuracy.

Table 5. Performance comparison of gait recognition models.

Methods	Accuracy	Recall	F1_Score	Memory size
CNN [31]	95.46%	95.52%	95.49%	13.1 MB
LSTM+CNN [32]	96.09%	95.87%	95.95%	14.7 MB
LSTM＆CNN [33]	95.22%	94.96%	95.22%	12.1 MB
MSCNN	96.89%	96.87%	96.91%	102 MB
MSDSCNN	97.98%	98.01%	97.98%	28.6 MB
MSDSCNN+BK	**98.15%**	**98.18%**	**98.16%**	**24.7 MB**

3.5.2. Performance optimization comparison of multiscale depthwise models

This section focuses on the optimization of multiscale depthwise networks. The RBs and CBAM attention mechanisms are added to the multiscale depthwise network in this paper, and Table 6 shows the results of the experiments in detail. Compared with the initial MSDSCNN, the network with the addition of the RB has 0.22% higher accuracy and 0.064 lower loss. This is due to the fact that the RB uses jump connections during the training process to realize that the input data can be propagated across layers and reduce the information missing, thus reducing the loss. The MSDSCNN, with the addition of the attention mechanism, improves the accuracy by 0.47% and reduces the loss by 0.071, compared with the initial network. This is because the CBAM effectively improves the network's ability to extract key features by extracting weights that act on different dimensions of the FM, thus improving the classification accuracy. The MSDSCNN network combining RBs and attention mechanism has the highest accuracy and the smallest loss, achieving the expected objective.

Table 6. Comparison of optimization results of multiscale depthwise models.

Methods	Accuracy	Loss
MSDSCNN	98.15%	0.1466
MSDSCNN+RB	98.37%	0.0824
MSDSCNN+CBAM	98.62%	0.0754
MSDSCNN+RB+CBAM	**99.24%**	**0.0393**

3.5.3. Comparison of multiple feature fusion models

This section will demonstrate the accuracy of the multiple-feature fusion model and compare several outstanding double-channel networks on the dataset of this study. To guarantee the uniformity of the experiments, RBs and attention mechanisms were added to all networks.

The convergence behavior during the training of this experiment is shown in Figure 11. There are four curves corresponding to the four training models. All these models were trained using the same dataset. The model proposed in this study has the fastest convergence speed, which is because this model can extract more complete and numerous features. When the number of extracted features increases, the training time is prolonged, but the basis of the scoring classification will increase, and the final scoring classification result will be higher.

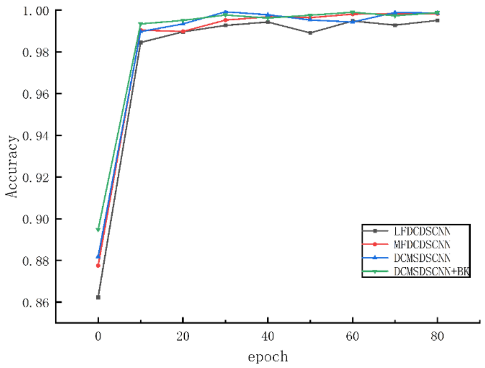


Figure 11. Comparison of the accuracy of various models under the same epoch curve.

Table 7. Performance comparison of multiple feature fusion models.

Methods	Accuracy	Loss	F1_Score	Memory size
LFDCDSCNN	98.16%	0.0985	98.16%	26.1 MB
MFDCDSCNN	98.18%	0.0966	98.14%	16.3 MB
DCMSCNN	98.37%	0.0942	98.33%	128 MB
DCMSDSCNN	99.34%	0.0312	99.30%	35.2 MB
DCMSDSCNN+BK	**99.60%**	**0.0289**	**99.62%**	**30.4 MB**

Table 7 shows the accuracy of all methods. Based on the same dataset, the experimental results show that the last fusion double-channel depthwise separable convolutional neural network (LFDCDSCNN) has the slowest convergence on the training set and the lowest final accuracy. The convergence of the model proposed in this paper is the best on the training set, and can achieve higher accuracy with a shorter training time. Middle fusion double-channel depthwise separable convolutional neural network (MFDCDSCNN) performs fusion after the second layer of CONV, while LFDCDSCNN performs fusion after the last layer of CONV. The comparison results display that the classification accuracy of MFDCDSCNN is slightly improved than LFDCDSCNN, but the memory size is 37.5% smaller. DCMSDSCNN is precisely based on MFDCDSCNN for improvement. The experimental results show that the DCMSDSCNN with the BK structure incorporated has the highest accuracy rate. Its classification accuracy is about 1.23% higher than that of the unoptimized double-channel multiscale convolutional neural network (DCMSCNN), and the memory footprint of the model is about 4.21 times lower. The accuracy is 0.26% higher than the DCMSDSCNN without the BK structure, and the memory footprint is reduced by 13.6%. The overall comparison shows that our model achieves better performance, providing higher classification accuracy at a little computational cost.

3.5.4. Comparison with recent studies

A total of 8210 samples were collected for five gait patterns simulated by 25 subjects, and LSTM-CNN [17] classified abnormal gait with 93.1% accuracy. The mixed RNN and CNN model [19] achieved 97.60% classification for normal gait, anesthetized gait, waddling gait, stepping gait, stiff-legged gait and Trendelenburg gait with accuracy (without considering the subjects), where 12 healthy subjects simulated the gait and 2880 data points were collected. zhao et al. [34] generated a lightweight OpenPose (OP) model by deep separable convolution to extract abnormal gait features, and used a machine learning algorithm for classification, which could reach 92.13% accuracy. Eight of the subjects participated in the experiment, simulating normal gait and five abnormal gait patterns. We collected 23910 samples simulated by 20 subjects, and achieved 99.60% classification accuracy for normal gait, fast gait, slow gait, hemiplegic gait and analgesic gait using a double-channel multiscale depthwise model.

Compared with other studies, we classified different gait patterns with more samples, and achieved better classification performance.

4. Conclusions

This study proposes a DCMSDSCNN-based method for identifying abnormal gaits. The method designs MDB, which is a module that utilizes different sizes of convolutional kernels corresponding to different sizes of perceptual fields to achieve multiscale feature extraction. The extracted multiscale features of the left and right legs are fused to achieve the fusion of the finer feature information to improve the performance of gait recognition. DSC is introduced to replace the standard convolution in MDB, and the amount of network parameters is significantly reduced by using the weighting operation of depthwise convolution and pointwise convolution. The application of BK structure to optimize the MDB further reduces the number of network parameters through dimensional changes, and achieves the computation cost reduction. The comparison experiments of the multi-feature fusion model show that the application of MDB and BK structure achieves the improvement of gait recognition accuracy while substantially reducing the computational cost. In future studies, we will work on converting the time series data into 2D images, varying the model inputs to verify the reliability of the model in various recognition scenarios, and applying the model in this paper to mobile devices as a way to test its potential in the real world.

Acknowledgments

We would like to thank all the colleagues that have supported this work. This work is jointly supported by Natural Science Foundation of Hebei Province (No. F2021201002); Science and Technology Project of Hebei Education Department (No. ZD2020146).

Conflict of interest

We declare that there are no conflicts of interest.

References

1. J. M. Hausdorff, D. A. Rios, H. K. Edelberg, Gait variability and fall risk in community-living older adults: A 1-year prospective study, *Arch. Phys. Med. Rehabil.*, **82** (2001), 1050–1056. https://doi.org/10.1053/apmr.2001.24893
2. S. Wu, J. Ou, L. Shu, G. Hu, Z. Song, X. Xu, et al., MhNet: Multi-scale spatio-temporal hierarchical network for real-time wearable fall risk assessment of the elderly, *Comput. Biol. Med.*, **144** (2022), 105355. https://doi.org/10.1016/j.compbiomed.2022.105355
3. I. Mileti, J. Taborri, S. Rossi, M. Petrarca, F. Patanè, P. Cappa, Evaluation of the effects on stride-to-stride variability and gait asymmetry in children with Cerebral Palsy wearing the WAKE-up ankle module, in *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, (2016), 1–6. https://doi.org/10.1109/MeMeA.2016.7533748
4. H. C. Chang, Y. L. Hsu, S. C. Yang, J. C. Lin, Z. H. Wu, A wearable inertial measurement system with complementary filter for gait analysis of patients with stroke or Parkinson’s disease, *IEEE Access,* **4** (2016), 8442–8453. https://doi.org/10.1109/ACCESS.2016.2633304
5. J. M. Hausdorff, M. E. Cudkowicz, R. Firtion, J. Y. Wei, A. L. Goldberger, Gait variability and basal ganglia disorders: stride-to-stride variations of gait cycle timing in Parkinson’s disease and Huntington’s disease, *Mov. Disord.*, **13** (1998), 428–437. https://doi.org/10.1002/mds.870130310
6. T. N. Nguyen, H. H. Huynh, J. Meunier, Skeleton-based abnormal gait detection, *Sensors*, **16** (2016), 1792. https://doi.org/10.3390/s16111792
7. R. Rucco, V. Agosti, F. Jacini, P. Sorrentino, P. Varriale, M. de Stefano, et al., Spatio-temporal and kinematic gait analysis in patients with Frontotemporal dementia and Alzheimer’s disease through 3D motion capture, *Gait Posture* **52** (2017), 312–317. https://doi.org/10.1016/j.gaitpost.2016.12.021
8. J. Jenkins, C. Ellis, Using ground reaction forces from gait analysis: Body mass as a weak biometric, in *International Conference on Pervasive Computing*, **4480** (2007), 251–267. https://doi.org/10.1007/978-3-540-72037-9_15
9. T. C. Pataky, T. Mu, K. Bosch, D. Rosenbaum, J. Y. Goulermas, Gait recognition: Highly unique dynamic plantar pressure patterns among 104 individuals, *J. R. Soc., Interface*, **9** (2012), 790–800. https://doi.org/10.1098/rsif.2011.0430
10. A. Mannini, D. Trojaniello, A. Cereatti, A. M. Sabatini, A machine learning framework for gait classification using inertial sensors: Application to elderly, post-stroke and huntington’s disease patients, *Sensors*, **16** (2016), 134. https://doi.org/10.3390/s16010134
11. M. Alaqtash, T. Sarkodie-Gyan, H. Yu, O. Fuentes, R. Brower, A. Abdelgawad, Automatic classification of pathological gait patterns using ground reaction forces and machine learning algorithms, in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (2011), 453–457. https://doi.org/10.1109/IEMBS.2011.6090063
12. N. Mezghani, S. Husse, K. Boivin, K. Turcot, R. Aissaoui, N. Hagemeister, et al., Automatic classification of asymptomatic and osteoarthritis knee gait patterns using kinematic data features

and the nearest neighbor classifier, *IEEE Trans. Biomed. Eng.*, **55** (2008), 1230–1232. https://doi.org/10.1109/TBME.2007.905388

13. G. He., M. Lin, Y. Ching, Methods for person recognition and abnormal gait detection using tri-axial accelerometer and gyroscope, in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, (2017), 1691–1694. https://doi.org/10.1109/CSCI.2017.294
14. Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50
15. I. Huitzil, L. Dranca, J. Bernad, F. Bobillo, Gait recognition using fuzzy ontologies and Kinect sensor data, *Int. J. Approx. Reason.*, **113** (2019), 354–371. https://doi.org/10.1016/j.ijar.2019.07.012
16. M. Gadaleta, L. Merelli, M. Rossi, Human authentication from ankle motion data using convolutional neural networks, in *2016 IEEE Statistical Signal Processing Workshop (SSP)*, (2016), 1–5. https://doi.org/10.1109/SSP.2016.7551815
17. J. Gao, P. Gu, Q. Ren, J. Zhang, X. Song, Abnormal gait recognition algorithm based on LSTM-CNN fusion network, *IEEE Access*, **7** (2019), 163180–163190. https://doi.org/10.1109/ACCESS.2019.2950254
18. J. Chakraborty, A. Nandy, Discrete wavelet transform based data representation in deep neural network for gait abnormality detection, *Biomed. Signal Process. Control*, **62** (2020), 102076. https://doi.org/10.1016/j.bspc.2020.102076
19. K. Jun, S. Lee, D. W. Lee, M. S. Kim, Deep learning-based multimodal abnormal gait classification using a 3D skeleton and plantar foot pressure, *IEEE Access*, **9** (2021), 161576–161589. https://doi.org/10.1109/ACCESS.2021.3131613
20. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
21. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 2818–2826. https://doi.org/10.1109/CVPR.2016.308
22. S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: Convolutional block attention module, in *European Conference on Computer Vision*, (2018), 3–19. https://doi.org/10.1007/978-3-030-01234-2_1
23. H. Huang, P. Zhou, Y. Li, F. Sun, A lightweight attention-based CNN model for efficient gait recognition with wearable IMU sensors, *Sensors*, **21** (2021), 2866. https://doi.org/10.3390/s21082866
24. K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in *European Conference on Computer Vision*, (2016), 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
25. M. Shafiq, Z. Gu, Deep residual learning for image recognition: A survey, *Appl. Sci.*, **12** (2022), 8972. https://doi.org/10.3390/app12188972
26. M. Paulich, M. Schepers, N. Rudigkeit, G. Bellusci, Xsens MTw Awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3D kinematic applications, *XSens Technol.*, (2018), 1–9. https://doi.org/10.13140/RG.2.2.23576.49929

27. M. Musci, D. De Martini, N. Blago, T. Facchinetti, M. Piastra, Online fall detection using recurrent neural networks, preprint, arXiv:1804.04976.
28. F. Chollet, Xception: Deep learning with depthwise separable convolutions, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 1800–1807. https://doi.org/10.1109/CVPR.2017.195
29. W. Li, X. Zhang, Y. Peng, M. Dong, Spatiotemporal fusion of remote sensing images using a convolutional neural network with attention and multiscale mechanisms, *Int. J. Remote Sens.*, **42** (2021), 1973–1993. https://doi.org/10.1080/01431161.2020.1809742
30. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *preprint*, arXiv:1412.6980.
31. A. Rohan, M. Rabah, T. Hosny, S. Kim, Human pose estimation-based real-time gait analysis using convolutional neural network, *IEEE Access*, **8** (2020), 191542–191550. https://doi.org/10.1109/ACCESS.2020.3030086
32. Q. Zou, Y. Wang, Q. Wang, Y. Zhao, Q. Li, Deep learning-based gait recognition using smartphones in the wild, *IEEE Trans. Inf. Forensics Secur.*, **15** (2020), 3197–3212. https://doi.org/10.1109/TIFS.2020.2985628
33. A. S. Alharthi, K. B. Ozanyan, Deep learning for ground reaction force data analysis: Application to wide-area floor sensing, in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, (2019), 1401–1406. https://doi.org/10.1109/ISIE.2019.8781511
34. Y. Zhao, J. Li, X. Wang, F. Liu, P. Shan, L. Li, et al., A lightweight pose sensing scheme for contactless abnormal gait behavior measurement, *Sensors*, **22** (2022), 4070. https://doi.org/10.3390/s22114070