SD-CNN: A Novel Lightweight Convolutional Neural Network Model for Fall Detection

Han-lin Shen[1], Tian-hu Wang[2]*, Hong Mu[3]
School of Electrical and Information Engineering, Jiangsu University of Technology, Changzhou, China [1, 2]
Changzhou Xutec System Technology Co., Changzhou, China[3]

***Abstract*—Aiming at the traditional deep learning fall detection model due to high computational complexity and a large number of parameters, this study proposes a lightweight convolutional neural network model, SD-CNN (SMA-Enhanced Depthwise Convolutional Neural Network), for fall detection. The model is first designed with an SMA attention module to enhance feature representation. Then, depth separable convolution is used to significantly reduce the model complexity. Finally, batch normalisation and Dropout regularisation techniques are combined to efficiently extract spatial-temporal features from temporal signals for accurate classification of fall and non-fall behaviours. The experiments use a sliding window to extract discrete features, three-axis acceleration, and synthetic acceleration as feature inputs. SD-CNN achieves 99.11% accuracy, 98.78% specificity, and 99.39% sensitivity on the homemade dataset Act, which are improved by 7.14%, 6.42%, and 9.38%, respectively, compared to CNN, while the number of parameters is reduced significantly. The effectiveness of the model is also verified by generalisation experiments on the public datasets SisFall and WEDAFall. The SD-CNN algorithm can efficiently complete the fall detection task, and the lightweight design makes it particularly suitable for wearable devices, which provides a highly efficient and reliable solution for real-time fall detection, and it has an important value for practical applications.**

Keywords*—Fall detection; lightweight; SMA attention; depth-separable convolution

I. Introduction

In the context of global ageing, falls are frequent and serious in the elderly. Not only is it the main cause of casualties, but it also often leads to hospitalisation and long-term rehabilitation, placing a heavy strain on society and the healthcare system [1]. Therefore, how to effectively detect and prevent fall events is of great relevance in the field of smart health monitoring [2].

Existing fall detection methods fall into three main categories: Machine vision-based inspection, Environmental sensor-based detection, and wearable device-based detection [3]. Machine vision-based detection relies on videos or images to determine behaviour. For example, Nana Vo et al. [4] used a lightweight network structure based on Yolov5s and OpenPose to identify video content with an accuracy of 95.43%. Mengdi Cao et al. [5] proposed a fall detection algorithm for complex scenes based on improved Yolov8, which uses video streams to identify fall behaviour in real-time, and validated on multiple video streaming datasets, achieving an average accuracy of 75.8%. However, such methods are limited in practical application by the detection range and device privacy, and cannot be widely spread. Detection based on environmental sensors relies on external sensors to obtain information about environmental changes. Wang Zhaojun et al. [6] designed an infrared array sensor system to identify human behaviour through temperature changes and used the KNN algorithm to achieve fall detection, with an accuracy rate of about 95.2%, but this approach is prone to receive interference from external environmental transformations. Detection based on wearable devices is more common. Liu Bo et al. [7] proposed a radial basis function (RBF) neural network to achieve fall detection using accelerometer data with 98.1% accuracy. Wei Jiaxue et al. [8] used an improved temporal convolutional network (TCN) to solve the problem of traditional RNN and CNN training models are complex and prone to gradient explosion phenomenon, the accuracy, sensitivity and specificity on the public and fusion datasets reached 99.43%, 98.70% and 99.34%, respectively, but the number of model parameters is relatively large, which is not conducive to the deployment on embedded devices.

To address this problem, this study proposes a lightweight fall detection model, SD-CNN, which employs Signal Modulating Attention (SMA) for processing time-series data collected by accelerometers, and which approximates the traditional self-attention mechanism [9] approach to capture long-time dependencies in the sequences through Random Feature Mapping (RFM). The use of depth-separable convolution significantly reduces the amount of computation, reduces the complexity of the feature extraction process, reduces the model parameters, and improves the computational speed, thus effectively applying to the task of fall detection.

*Corresponding author.

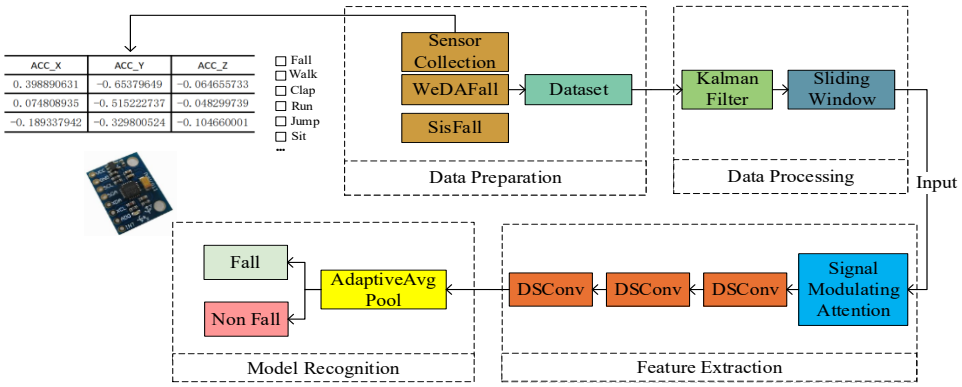

Fig. 1. General flow of the SD-CNN fall detection algorithm.

II. Related Work

A. SD-CNN Model

The key points of building a fall detection model are the appropriateness of feature extraction and the reasonableness of the constructed network model. The traditional convolutional network model is widely recognised for dealing with image problems, but due to the limitation of its convolutional kernel size, deeper networks are usually required to achieve more satisfactory results when dealing with long time series data. To address such problems, this study successively introduces the SMA attention mechanism and depth-separable convolution. SMA attention can achieve efficient and accurate fall detection on long time series data by optimising feature extraction and reducing the number of input channels. Deep separable convolution extracts feature points by decomposing the standard convolution into two independent operations, which significantly reduces the amount of computation and the number of parameters while extracting local features in each channel. The SD-CNN model consists of three depth-separable convolutional blocks, each containing a channel-by-channel convolutional layer and a point-by-point convolutional layer. The convolution kernel sizes are all set to 3, with a step size of 1 and padding of 1 to keep the time series length consistent. The number of output channels for the three convolution blocks is 32, 32, and 64 in that order. Each convolutional block is followed by batch normalisation, ReLU activation function, and average pooling (kernel size 2, step size 2) in order to progressively compress the feature length and improve feature stability. In terms of input feature design, this study divides the three-axis acceleration sensor data into fixed-length segments in a sliding-window fashion, with each sample containing 120 time steps, three channels (x, y, and z direction acceleration), and the final input dimensions of the model are 3×128. The fall detection task mainly consists of four parts, namely, data acquisition, data preprocessing, feature extraction, and model inference, and the overall layout is shown in Fig. 1.

B. SMA Attention

Although the traditional dot product attention mechanism can highlight important features of the input data and suppress unimportant features to improve the efficiency and accuracy of feature extraction, it has a computational complexity of $O(n^2)$, and the memory requirement grows by a square as the length of the sequence increases, which is particularly unfavourable for deployment in embedded devices. Therefore, in this study, we design the SMA attention mechanism, which combines the RFM idea of the EA attention mechanism [10] to reduce the computational complexity to O(n). RFM essentially provides an efficient form of approximate integration to capture long-range dependencies, where the contribution of each time step is accumulated in the global attention score. The A_{raw} obtained after normalisation characterises the distribution of importance of each time step feature in the global context, i.e, Eq. (1):

$$A_{raw} = \sum_{l=1}^{L} Q_{rf}[:, l, :] \odot K_{rf}[:, l, :] \quad (1)$$

where, the contribution of each time step is accumulated in the global attention score. A_{raw} obtained after normalisation characterises the distribution of the importance of each time step feature in the global context; larger A_{raw} correspond to signals with a consistently high response over multiple time steps (indicating that the feature is stable and important over time in the time series); smaller A_{raw} correspond to noise or short-term fluctuations (automatically suppressed by the model). The structure of the SMA attention mechanism is shown in Fig. 2.

For input feature mapping $X \in \mathbb{R}^{b\times C\times L}$, Channel dimensionality reduction is first performed by two 1×1 convolutions [Eq. (2) and Eq. (3)]:

$$Q = \mathrm{Conv1d_q}\mathbb{R}^{b\times C_r\times L} \quad Q \in \mathbb{R}^{b\times C_r\times L} \quad (2)$$

$$K = \mathrm{Conv1d_k}\mathbb{R}^{b\times C_r\times L} \quad K \in \mathbb{R}^{b\times C_r\times L} \quad (3)$$

where, b is the batch size, C is the number of input channels, L is the length of the sequence, Cr is the number of channels after dimensionality reduction, and r is the ratio of reduction.

Then, Q^T and K^T are projected into a random feature space using a random feature matrix $R_F \in \mathbb{R}^{C_r\times F}$, and ReLU activation is applied, resulting in Eq. (4) and Eq. (5):

$$Q_{rf} = \mathrm{ReLU}(Q^T \odot R_F) \in \mathbb{R}^{b\times L\times F} \quad (4)$$

$$K_{rf} = \mathrm{ReLU}(K^T \odot R_F) \in \mathbb{R}^{b\times L\times F} \quad (5)$$

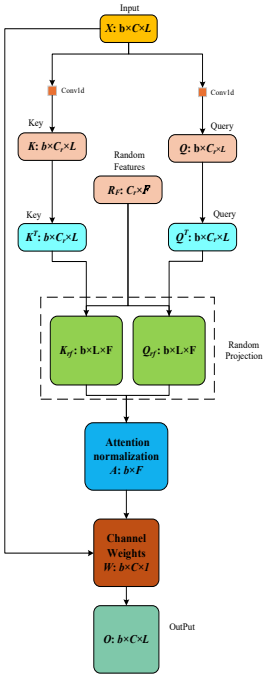

Fig. 2. Structure of the SMA attention mechanism.

where, Q^T and K^T denote the transposition of the channel dimension and the sequence length dimension, F is the number of random features. In the random feature space, Attention scores A_{raw} and normalisation factor N are then calculated as follows [Eq. (6) and Eq. (7)]:

$$A_{raw} = \sum_{l=1}^{L} Q_{rf}[:, l, :] \odot K_{rf}[:, l, :] \in \mathbb{R}^{b \times F} \quad (6)$$

$$N = \sum_{l=1}^{L} K_{rf}[:, l, :] \odot K_{rf}[:, l, :] \in \mathbb{R}^{b \times F} \quad (7)$$

where, $\odot$ is element-by-element multiplication. Then find the normalised attention fraction A, as in Eq. (8):

$$A = \frac{A_{raw}}{N+\varepsilon} \in \mathbb{R}^{b \times F} \quad (8)$$

where, ε is a small constant for numerical stability. The attention scores are further extended to all channels and averaged over the feature dimensions [Eq. (9)]:

$$W = \frac{1}{F} \sum_{f=1}^{F} A[:, f] \in \mathbb{R}^{b \times C \times 1} \quad (9)$$

Finally, the calculated weights are applied directly to the original input, as in Eq. (10):

$$O = X \odot W \in \mathbb{R}^{b \times C \times L} \quad (10)$$

C. Depthwise Separable Convolution

Depthwise Separable Convolution (DSC) consists of depth-by-depth convolution and point-by-point convolution, where depth-by-depth convolution is used to extract spatial features and point-by-point convolution is used to extract channel features [11]. Depth-separable convolution groups convolutions in feature dimensions, perform independent depth-by-depth convolution for each channel, and aggregate all channels using a 1x1 convolution (pointwise convolution) before output. This approach reduces the convolutional dimension and can effectively reduce the convolutional kernel parameters and unnecessary computations. The DSC convolutional network structure is shown in Fig. 3.

The input feature map size in Fig. 3 is DE×DF, the number of input channels is M, the size of the convolution kernel is DK, and the number of output channels is N. If the same number of channels is output using an ordinary convolutional product network, the number of parameters is DK×DK×M×N, whereas when using a depth-separable convolutional network, the number of parameters is DK×DK×M+M×N, The number of parameters is only $\frac{1}{N} + \frac{1}{D_k^2}$ times that of a normal convolution. As DK and N increase, the number of covariates of DSC relative to conventional CNN decreases dramatically.

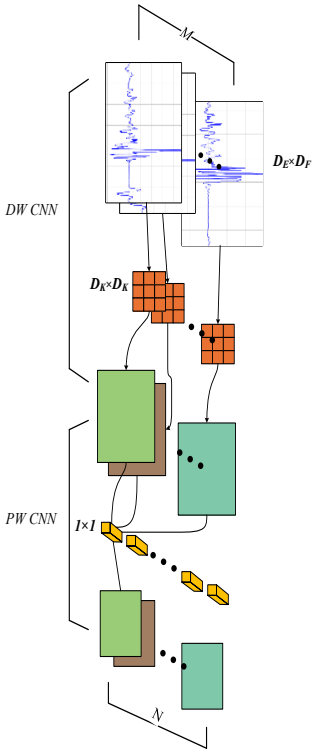

Fig. 3. Depth-separable convolutional network structure.

III. Experimental Data and Analysis

A. Datasets and Experimental Environments

There are three datasets in this study. The first one is the homemade dataset Act, and the data in the Act dataset is collected by the mpu6050 sensor. The stm32f103c8t6 is used as the main control chip, and the data sampling frequency is 50Hz. The collection was completed by 12 volunteers, who consisted of 12 young people on whom the sensor was mounted on the left wrist. The dataset contains 5 fall activities (falling backwards, falling forwards and landing on the knee, falling forwards, failing to get up from a sedentary position resulting in a fall, and falling from a standing position) and nine non-falling activities (walking, jogging, sitting down, trotting, going up the stairs, going down the stairs, opening and closing doors, applauding, and jumping), a total of 4,200 data

items are included. The dataset training set, test set and validation set are divided in the ratio of 8:1:1.

The remaining two datasets are the public datasets WEDAFall and SisFall [12]. The WEDAFall dataset equipment samples primarily at 50Hz, and data collection was done by 31 volunteers, the volunteers consisting of 20 young people and 11 older people, collected data via sensors placed on their wrists and contained a total of 3,719 pieces of data on 8 fall activities and 11 non-fall activities. The SisFall dataset device had a sampling frequency of 200Hz, and data collection was done by 38 volunteers, consisting of 23 young people and 15 older people, who collected data via sensors placed on their waists, containing a total of 4,510 pieces of data, capturing 15 fall activities and 19 non-fall activities. The specific data distribution is shown in Table I.

The deep learning framework for the experiments in this study is Pytorch, the training environment is a 64-bit Windows 11 operating system, the CPU is i7-12650H, the graphics card uses RTX-4060, the RAM is 16GB, and the code running environment is Python 3.8.

TABLE I. DATASET INFORMATION

	Non-Fall	Fall
Act	2000	2200
WEDAFall	2319	1400
Sisfall	2935	1575

B. Performance Indicators

For fall detection, the accurate identification of fall events is very important, and there are three evaluation metrics used in this study, which are accuracy, sensitivity, and specificity. Accuracy is an important measure of the model's overall classification ability, indicating the number of samples correctly predicted by the model as a proportion of the total number of samples. The equation is as follows [Eq. (11)]:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

where, TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

Sensitivity and specificity, on the other hand, respond to the model's ability to distinguish between positive and negative samples. The equation is as follows [Eq. (12) and Eq. (13)]:

$$SP = \frac{TN}{TP+FN} \tag{12}$$

$$SE = \frac{TP}{TN+FP} \tag{13}$$

where, SP is specificity and SE is sensitivity, specificity and sensitivity are complementary to each other, the higher the sensitivity indicates that the model has a higher recognition rate for fall events, and the two together measure the model's ability to discriminate on positive and negative class samples.

C. Data Preprocessing

The raw triaxial acceleration components (*ax, ay, az*) are projections to the sensor's own coordinate axes and are highly dependent on its spatial attitude. Using this data directly, the model has to learn not only the motion itself, but also the complex relationship between the motion patterns and the changing sensor poses. This double learning burden increases model complexity and may impair generalisation capabilities, especially in conditions where sensor poses are unknown or changing. According to vector theory, the modulus of a vector does not change with the rotation of the coordinate system [13]. In other words, no matter how the sensor is rotated, the calculated acceleration magnitude S is constant as long as the magnitude of the total acceleration it senses remains constant.

Therefore, this study uses the acceleration amplitude as the feature, and this approach can effectively eliminate the interference caused by the sensor attitude change, making the feature robust to the sensor installation position and orientation. Acceleration amplitude indicates the size of the acceleration. The amplitude of the acceleration of the human body movement state reflects the intensity of the human body movement, which is defined as shown in Eq. (14):

$$S = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{14}$$

1) Sliding window algorithm: To extract the temporal characteristics of the data, a sliding window strategy is used. For dynamic behaviours with long duration, a larger window allows for complete capture of their signals [14]. However, blindly increasing the window size is not optimal and may lead to increased computational complexity or failure to accurately capture behavioural changes at shorter times. Therefore, a reasonable choice of window size is essential to balance the completeness of timing information with computational efficiency.

A complete fall consists of three main phases: the imbalance phase, the falling phase, and the contact with the ground phase, as shown in Fig. 4. The duration of these phases and is approximately 2s to 3s, respectively, in order to unify the size of the input data, this study makes the window able to cover a complete information of the fall event by setting the window size to 120 and the window sliding step to 10, and splits the time series into multiple fixed-length sub-sequences through the window sliding, and this method overcomes the situation that different behaviours may span different time periods, and at the same time, the model provides a sufficiently large number of training samples, which makes full use of the time series data characteristics and enables the model to capture long-time dependencies.

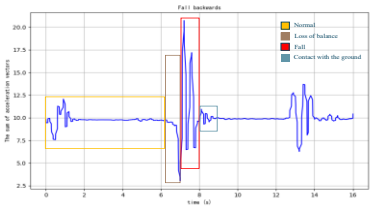

Fig. 4. Analysis of backward falling activities.

2) Low-pass filtering: Accelerometer sensors are prone to receive external noise interference when collecting data, and it

can be seen through spectral analysis that accelerometer noise is usually high-frequency noise. The frequency of human motion signals is low, with the frequency range of daily activities such as walking and running typically ranging from 0.5 Hz to 5 Hz, and the frequency of instantaneous acceleration changes for falls typically ranging from 1 Hz to 20 Hz, as shown in Fig. 5.

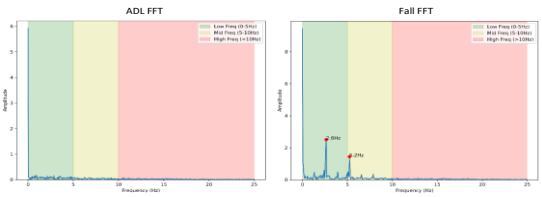

Fig. 5. Spectral analysis of human activity FFT plot.

In this study, a Butterworth low-pass filter is used to remove high-frequency noise from acceleration sensor data, which eliminates the need to rely on a physical model of signal generation and effectively reduces the interference of noise with subsequent detection tasks. After the cut-off frequency is processed, the data becomes smoother and retains the data features to have a better performance in training, as shown in Fig. 6.

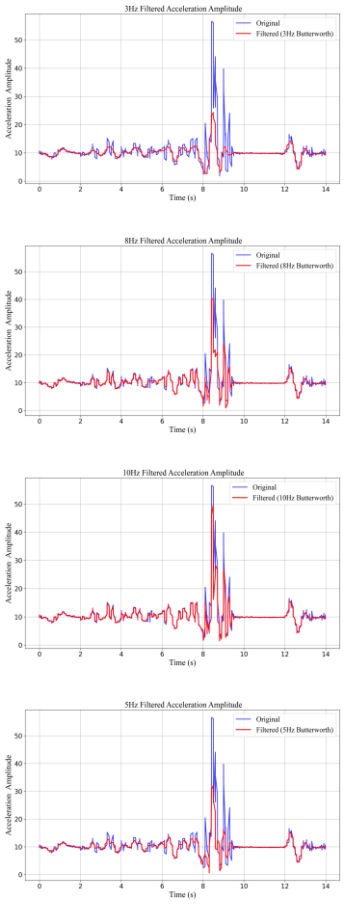

Fig. 6. Comparison of the results of Butterworth low-pass filter processing.

After comparative experiments, it was found that the model was best trained when the cut-off frequency was set to 5.5 Hz, effectively removing the interference from the sensor (see Table II).

TABLE II. COMPARISON OF TEST RESULTS

Filtering Frequency	Acc%	SP/%	SE%
\	96.34	96.25	97.11
2.5Hz	98.31	97.26	98.78
5.5Hz	**99.11**	**98.38**	**99.39**
8Hz	98.42	98.21	99.01
10Hz	97.73	96.94	98.34

IV. RESULTS

A. Comparative Experiments on Correlation Models

In order to comprehensively evaluate the performance of the proposed model and verify its effectiveness in fall detection tasks, a series of comparative experiments is designed in this study. Traditional detection methods and several state-of-the-art deep learning models were selected, including Threshold Analysis (Threshold) [15], Artificial Neural Network (ANN) [16], Convolutional Neural Network (CNN) [17], Temporal Convolutional Network (TCN) [8], Long Short-Term Memory Networks (LSTMs) [19], Gated Recurrent Neural Network (GRU) [18] and Bidirectional Long Short-Term Memory Network (Bi-LSTM) [19]. They are trained and tested on the same dataset and experimental conditions. These models were chosen because of their wide range of applications in processing time-series data and feature extraction. Through experimental comparisons, the SD-CNN model proposed in this study improves the accuracy, specificity and sensitivity by 15.31%, 16.29% and 9.38%, respectively, compared with the traditional detection method, Threshold. Compared with several other state-of-the-art deep learning models, the model in this study has the smallest number of parameters while maintaining the highest accuracy, specificity and sensitivity. This is a good indication that the SD-CNN model performs more superior in terms of classification accuracy and robustness. The specific data are shown in Table III.

TABLE III. COMPARISON OF MODELS

	Acc%	SP/%	SE%	Params	FLOPs
Threshold	83.80	89.00	83.10	/	
CNN	91.97	92.36	90.01	28480	14.6M
LSTM	93.94	92.44	95.73	46754	37.8M
GRU	95.70	87.50	96.80	47856	38.2M
Bi-LSTM	96.73	99.33	95.26	85866	42.5M
TCN	98.43	99.13	98.27	197,120	39.7M
ANN	94.02	93.20	94.21	271330	124.4M
SD-CNN	**99.11**	**98.78**	**99.39**	4076	0.97M

B. Cross-Dataset Evaluation

In order to comprehensively assess the effectiveness and usefulness of the proposed model in this study, avoid overfitting of the model to specific training data, and validate its generalisation ability on different data sources, generalisation experiments across datasets are designed and implemented in this section. In this study, two public datasets, SisFall and WEDAFall, which are widely used in the field of fall detection, are selected for testing, aiming to examine the performance of the models when confronted with new data that is different from the source of the training data. The experimental environment is consistent with "3.1" of this study. As can be seen from Table IV, SD-CNN improves accuracy, specificity, and sensitivity by 6.92%, 6.23% and 7.34%, respectively, over CNN on the WEDAFall dataset. On the SisFall dataset, the accuracy, specificity, and sensitivity are improved by 6.38%, 8.89% and 8.42%, respectively, over CNN.

TABLE IV. GENERALISATION EXPERIMENTS

Models	WEDAFall			SisFall		
	Acc%	*SP/%*	*SE%*	*Acc%*	*SP/%*	*SE%*
CNN	91.34	91.98	90.44	92.78	89.65	90.18
SD-CNN	**98.26**	**98.21**	**97.78**	**99.16**	**98.54**	**98.60**

V. DISCUSSION

The SD-CNN neural network architecture proposed in this study aims to achieve efficient and accurate human activity recognition, especially fall detection. The model achieves significant lightweighting at the model structure level by integrating deeply separable convolutions to reduce the computational effort and incorporating the SMA attention mechanism to dynamically focus on key features. The experimental evaluation results show that SD-CNN not only has significant advantages in terms of the number of model parameters and the expected computational resource consumption, but also outperforms other models in terms of key performance indicators, including 99.11% accuracy, 98.78% specificity, and 99.39% sensitivity. This proves the effectiveness of the lightweight architecture proposed in this study. Nevertheless, this study still has some limitations. Although SD-CNN shows good computational efficiency on resource-constrained devices, it still needs to be further optimised in scenarios with extreme low power consumption or higher real-time requirements. Second, the design of the attention mechanism in this study is still a relatively simplified and approximate form, and its performance differences in multi-channel and multi-modal data have not yet been fully explored.

VI. CONCLUSION

The main contribution of this research is to provide a lightweight activity recognition model, SD-CNN that balances efficiency, performance, and robustness, which is capable of recognising human activities effectively, especially in fall detection tasks. In the future, we will continue to explore this model by extending it to multimodal sensor data fusion, investigating more efficient attentional mechanisms, and validating it in larger and more diverse groups of people and scenarios to further advance its translation into practical applications.

ACKNOWLEDGMENT

This work is one of the phased achievements of Jiangsu Province Postgraduate Practical Innovation Project (SJCX24_1796), Changzhou Science and Technology Project (CZ20230025, CZ20250010, CJ20241070), and Jiangsu University of Science and Technology Postgraduate Practical Innovation Plan Project (XSJCX_46).

REFERENCES

[1] K. Durga Bhavani and M. Ferni Ukrit, "Design of inception with deep convolutional neural network based fall detection and classification model," Multimedia Tools and Applications, vol. 83, pp. 23799–23817, 2024.

[2] S. K. Gharghan and H. A. Hashim, "A comprehensive review of elderly fall detection using wireless communication and artificial intelligence techniques," Measurement, vol. 226, Art. no. 114186, 2024.

[3] Y. Li, P. Liu, Y. Fang, et al., "A Decade of Progress in Wearable Sensors for Fall Detection (2015–2024): A Network-Based Visualization Review," Sensors, vol. 25, p. 2205, 2025.

[4] F. Nana, L. Daming, C. Xiaoting, et al., "A fall detection algorithm based on the lightweight OpenPose model," Sensors and Microsystems, vol. 40, pp. 131–134, 2021.

[5] C. Mengdi, Y. Mengfan, W. Liuyi, et al., "A fall detection algorithm for complex scenes based on improved YOLOv8," Modern Information Technology, vol. 9, pp. 66–71, 2025.

[6] Z. Wang, Z. Xu, and L. Chen, "Research on Human Behaviour Recognition System Based on Infrared Array Sensor," Infrared Technology, vol. 42, pp. 231–237, 2020.

[7] B. Liu, W. Kong, J. Xiao, and M. Wang, "Fall detection algorithm based on RBF neural network," Computer Technology and Development, vol. 32, pp. 167–172, 2022.

[8] J. Wang, G. Gao, and G. Teng, "Application of improved TCN algorithm for human fall detection," Computer Engineering and Design, vol. 44, pp. 2859–2866, 2023.

[9] C. Li, M. Liu, X. Yan, et al., "Research on CNN-BiLSTM fall detection algorithm based on improved attention mechanism," Applied Sciences, vol. 12, p. 9671, 2022.

[10] Z. Shen, M. Zhang, H. Zhao, et al., "Efficient attention: Attention with linear complexities," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., pp. 3531–3539, 2021.

[11] R. Liqiang, H. Wang, X. Peng, et al., "Complex manoeuvre recognition based on wavelet time-frequency maps and lightweight CNN-Transformer hybrid neural networks," J. Beijing Univ. Aeronaut. Astronaut., pp. 1–24, 2025.

[12] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "SisFall: A fall and movement dataset," Sensors, vol. 17, no. 1, p. 198, 2017.

[13] J. Handong and J. Wengang, "Simulation Study of Attitude Solving Based on Multi-MEMS Sensor Combination," Instrumentation Technology and Sensors, vol. 8, pp. 81–85, 2020.

[14] L. Xiaoguang, J. Shaokang, W. Zihui, et al., "Real-time fall detection method based on threshold and extreme random tree," Computer Application, vol. 41, pp. 2761–2766, 2021.

[15] T. Wang, B. Wang, Y. Shen, et al., "Accelerometer-based human fall detection using sparrow search algorithm and back propagation neural network," Measurement, vol. 204, p. 112104, 2022.

[16] K. Lisa, W. Suzheng, Y.-Q. Chen, et al., "A Review of Fall Detection Algorithms Based on Wearable Devices," J. Zhejiang Univ., vol. 52, pp. 1717–1728, 2018.

[17] L. Yan, Z. Meng, J. Wuhào, et al., "Design of fall detection system for the elderly using convolutional neural network," J. Zhejiang Univ., vol. 53, pp. 1130–1138, 2019.

[18] F. Luna-Perejón, M. J. Domínguez-Morales, and A. Civit-Balcells, "Wearable fall detector using recurrent neural networks," Sensors, vol. 19, no. 22, p. 4885, 2019.

[19] M. Dong and J. Peng, "A study on wearable fall detection based on bidirectional LSTM neural network," J. Guangxi Normal Univ., vol. 40, pp. 141–150, 2022.