A Video-Based Fall Detection Using 3D Sparse Convolutional Neural Network in Elderly Care Services

Fangping Fu
School of Medicine and Health, Shunde Polytechnic, Foshan, China
fangpingsd@outlook.com

Submitted: 31 Jul 2024 **Accepted:** 15 Jan 2025 **Published:** 28 Mar 2025

Abstract Falls in the elderly have become one of the major risks for the growing elderly population. Therefore, the application of automatic fall detection system for the elderly is particularly important. In recent years, a large number of deep learning methods (such as CNN) have been applied to such research. This paper proposed a sparse convolution method 3D Sparse Convolutions and the corresponding 3D Sparse Convolutional Neural Network (3D-SCNN), which can achieve faster convolution at the approximate accuracy, thereby reducing computational complexity while maintaining high accuracy in video analysis and fall detection task. Additionally, the preprocessing stage involves a dynamic key frame selection method, using the jitter buffers to adjust frame selection based on current network conditions and buffer state. To ensure feature continuity, overlapping cubes of selected frames are intentionally employed, with dynamic resizing to adapt to network dynamics and buffer states. Experiments are conducted on Multi-camera fall dataset and UR fall dataset, and the results show that its accuracy exceeds the three compared methods, and outperforms the traditional 3D-CNN methods in both accuracy and losses.

Keywords: 3D convolutional neural network, sparse convolution, fall detection, jitter buffer.

1. Introduction

Research shows that as the global population ages, more and more elderly people choose or have to live alone. The global population of elderly people living alone is growing, which has become a social phenomenon that has attracted much attention. In some developing countries, providing a care support framework system for elderly people living alone is still a long-term project [20]. Falls in the elderly have become one of the major risks for the growing elderly population. Due to the frailty of the elderly and some underlying diseases, they often fall, which has a serious impact on their health. Even a minor fall may cause fractures, broken bones or soft tissue injuries that cannot be fully healed. According to research, falls are one of the main causes of direct and indirect death among the elderly [11].

Monitoring systems often face the challenge of integrating advanced recognition algorithms. They often systematically utilize computer vision, digital image processing, vectorization, artificial intelligence (AI) and multithreading for tasks such as feature recognition. In recent years, people have shown great interest in the processing of 3D videos. People often try to make in-depth and multi-dimensional descriptions of 3D videos. Many existing fall detection systems rely on traditional image processing techniques or deep learning methods, but most methods still have problems such as high computational complexity, poor real-time performance, or inability to adapt to diverse scenarios. For example, the application of traditional 3D-CNN in video data processing often faces high computational load, especially in systems that require real-time or near real-time response, where computational efficiency becomes a bottleneck. Reference [15] proposed an automatic recognition system that uses gait analysis to identify individuals at a distance and predict the possibility of matching between gait profiles. In recent years, people have shown great interest in the processing of 3D videos. People often try to describe 3D videos in depth and in multiple dimensions. For example, [16] proposed a numerical analysis of stochastic differential equations describing body sway based on 3D video clips. 3D-CNN is a neural network that describes the 3D nature of videos in general. In this article, we integrate a simplified 3D-SCNN. For action recognition and behavior analysis, 3D-CNN can learn complex action patterns by analyzing the changes in multiple consecutive frames in the video. This ability is particularly important in action recognition, motion prediction and behavior analysis. It can be used to detect the falling posture of the elderly, and compared with traditional 3D-CNN, it has less computational complexity, so it is suitable for semi-real-time warning functions.

This paper focuses on the developed system based on SIP [26] and GB/T28181 [13] and the method of fall detection in community elderly care. Cameras can provide very rich information about people and the environment, and their presence is becoming more and more important in many daily environments due to the necessity of monitoring. Especially in community elderly care centers, the installation of cameras is particularly important. Because, given the limited staff in community elderly care centers, reliable fall detection systems based on video monitoring may play a very important role in future health care and assistance systems.

As research results, the main contributions of this paper are as follows:

(1) To the best of our knowledge, this is the first time that a sparse convolution operation is proposed, which can be applied to 2D convolution and 3D convolution. In 3D-SCNN, all convolution layers use this sparse convolution operation. In this sense, reducing the computational complexity of convolution operations without losing accuracy is crucial for accelerating video analysis and classification. Compared with traditional 3D-CNN, sparse convolution not only achieves efficient computation in the spatial domain, but also reduces memory usage, allowing video analysis tasks to run under low computing resources, adapting to real-time or semi-real-time monitoring needs, especially suitable for time-sensitive scenarios such as falls of the elderly.

(2) In the preprocessing stage, we use a dynamic adjustment method to select key frames. When using the jitter buffer to select key frames, the key frame selection strategy is mainly dynamically adjusted according to the current network status and the state of the buffer. In order to ensure the continuity of features, we deliberately overlap the

cube composed of multiple frames selected each time, so as to ensure the integrity and continuity of the convolution features. The overlapping part will also be dynamically resized according to the network status and the state of the buffer. Through dynamic adjustment, based on the current network status and buffer status, the selection of key frames and the processing of overlapping areas are optimized to ensure the continuity and integrity of feature information, avoiding the problem of too many or too few key frames in traditional methods, thereby improving the stability and accuracy of the system.

Finally, this study has strong practical application value. The system proposed in this paper can be applied to community nursing homes, home care and other fields to provide more efficient safety monitoring for the elderly and improve the quality of life of the elderly. With the continuous development of camera technology and computer vision algorithms, the fall detection system based on video surveillance will become an important part of future smart elderly care. In addition to fall detection, the system based on video surveillance can also be extended to other areas of elderly health monitoring, such as real-time monitoring of heart rate, respiratory rate, abnormal movements, etc., to provide comprehensive support for the health management of the elderly.

The rest of the article is structured as follows: Section 2 provides a review of the literature, covering several CNN-based methods relevant to our work. Section 3 outlines our video monitoring and analysis platform using the SIP protocol and GB/T28181. Section 4 illustrates the proposed 3D-SCNN approach to solve the Fall Detection case. Experimental validations of the proposed method are detailed in Section 5. Finally, Section 7 concludes the article and discusses the future research.

2. Related works

Convolutional neural network (CNN) is a particularly effective deep learning model. By simulating the processing method of biological visual systems, it can automatically extract features from data and build more complex feature representations layer by layer. This structure makes CNN particularly outstanding in tasks such as image recognition, target detection and classification. Reference [35] proves that the ten-fold cross-validation accuracy and recognition time of music emotion recognition under the CNN method are better than those of support vector machine (SVM) and Bayesian model. Reference [36] proposes a pupil and infrared spot detection method based on CNN to overcome the poor robustness of traditional eye tracking algorithms.

Current research on human fall detection is generally divided into two approaches. The first involves wearable sensors. Although these sensors have high accuracy and are computationally cheaper, their limitation is that they are highly invasive and the elderly are either unwilling to wear them or forget to wear them. The second technology involves computer vision, including machine learning and deep learning methods, which have high accuracy and robustness, strong versatility, and are less invasive and suitable for deployment.

Since video is a continuous arrangement of consecutive frames or images, it creates a smooth and continuous impression for the viewer. In deep learning, CNN is widely used in the video field due to its strong computing power and high accuracy. Typical applications include video analysis and classification, human pose estimation, and human detection. For example, Núñez-Marcos et al. [23] used optical flow images as input to the CNN network in order to get rid of the influence of environmental features. These optical flow images only represent the motion of consecutive video frames and ignore any appearance-related information such as color, brightness, or contrast.

3D Convolutional Neural Networks (3D-CNN) can effectively obtain feature representations from images and can exploit temporal and spatial details in the same convolution without being significantly affected by image processing. Due to these advantages, 3D-CNN has also become one of the hot spots in CNN research in recent years. The following are several common 3D-CNN networks: a) C3D (Convolutional 3D) [32] is a classic 3D convolutional neural network, originally proposed by Tran et al. in 2014. It performs well in tasks such as video action recognition, using 3D convolutional layers to capture spatiotemporal features. b) I3D (Inflated 3D ConvNet) [6] is a network based on 2D convolutional network (such as Inception architecture) extended to 3D space. It initializes the 3D model by extracting parameters from the 2D pre-trained model, making training more efficient. c) R(2+1)D (Residual 2+1D Networks) [33] is a deep residual network that combines 2D convolution and 1D temporal convolution. It improves efficiency and performance by decomposing space and time into 2D and 1D processing units. d) SlowFast Networks [10] is an architecture proposed by Feichtenhofer et al. to resolve the time scale difference between fast and slow motion in videos. It contains two streams (Slow and Fast), each stream uses a different frame rate to process the video. e) P3D (Pseudo-3D Networks) [24] network processes spatiotemporal information through 3D convolution, but the size and number of its convolution kernels are limited to 2D convolution to reduce computational costs.

In recent years, the research on fall detection based on 3D-CNN has shown an upward trend. Most of the research focuses on: the fusion of multi-stream technology, the use of autoencoders, fusion with LSTM, and methods and practices based on skeleton technology.

In research of multi-stream technology, Alanazi et al. [2] proposed a human fall detection system using a fused multi-stream 3D CNN, which corresponds each stream to one of the four stages of human fall (standing or walking, falling, falling, and stationary). Alanazi et al. [1] used an innovative 4-stream 3D convolutional neural network (4S-3DCNN) model to learn different but continuous spatial and temporal features. The system processes video input or real-time surveillance, uses a fine-tuned deep learning model to segment the presence of human body every 32 frames, and applies three-level image fusion to highlight motion differences. The technology generates four pre-processed images, which are input to the 4S-3DCNN model for classification. Continuous detection of "fall" actions triggers an alarm for immediate intervention.

In aspect of autoencoders research, Nogas et al. [21] proposed a new framework DeepFall, which proposed a new use of deep spatiotemporal convolutional autoencoders to learn spatial and temporal features from normal activities using non-intrusive perception patterns. The proposal method in [31] is based on a 3D fully convolutional neural network, namely 3DFCNN, which automatically encodes spatiotemporal patterns in the original depth sequence. The described 3D-CNN allows the classification of actions based on the spatial and temporal encoding information of the depth sequence. The proposed 3DFCNN is optimized to achieve good performance in terms of accuracy while working in real time. The paper [27] proposes a new method to improve fall detection in thermal image data using stacked AutoEncoder (AE) and 3-D convolutional neural network (3D-CNN) models, which are input into a meta-neural network that is trained to detect falls and non-falls.

About importing with LSTM, Reference [30] proposed a convolutional neural network long short-term memory model (1D CNN LSTM) for automatic recognition of robot behavior. It extracts the features of the robot task from a one-dimensional convolutional layer, followed by a recurrent layer to retrieve temporal information from the data.

Su et al. [29] proposed the three-dimensional convolutional neural network (3D-CNN) and fully connected long short-term memory network (FC-LSTM) have been shown to be a powerful non-invasive fall detection method. A new model combining lightweight 3D-CNN and convolutional long short-term memory (ConvLSTM) network is proposed. Channel and spatial attention modules are adopted in each layer to improve the detection performance. In addition, ConvLSTM is proposed to extract long-term spatiotemporal features of 3D tensors.

Using the skeleton method, Jayaswal et al. [12] introduced a novel approach for indoor fall detection using 3D convolutional neural networks (3D-CNNs) with temporal attention mechanisms. The pose estimation method is to analyze the coordinates of skeletal interest points and extract relevant features for accurate fall detection. The fusion of deep learning-based video analysis and pose estimation promotes the improvement of a powerful fall detection framework. Xiong et al. [34] proposed a skeleton-based 3D continuous low pooling neural network (S3D-CNN) for fall detection. In S3D-CNN, an active feature clustering selector is designed to extract skeleton representations from deep videos using a pose estimation algorithm and form optimized skeleton sequences for fall periods. A 3D continuous low pooling (3D-CLP) neural network is proposed to process these representation sequences by improving the number of network layers, pooling kernel size, and single input frame number. Noor et al. [22] proposed a new enhanced human pose dataset to improve the accuracy of pose extraction. They proposed

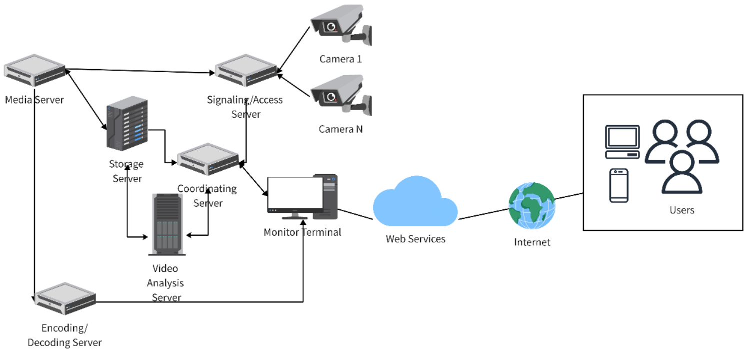


Fig. 1. Video surveillance and analysis platform system architecture.

a lightweight skeleton-based 3D-CNN behavior recognition network. Experimental results show that the proposed skeleton-based method shows high accuracy and efficiency in real-world scenarios. Xiong et al. [34] proposed a skeleton-based three-dimensional continuous low pooling neural network (S3D-CNN) for fall detection. In S3D-CNN, an activity feature clustering selector is designed to extract skeleton representations from depth videos using a posture estimation algorithm and form an optimized skeleton sequence of fall periods. A three-dimensional continuous low pooling (3D-CLP) neural network is proposed to process these representation sequences by improving the number of network layers, pooling kernel size, and single input frame number.

In summary, the research on 3D convolution operation and fall detection suitable for fast processing of surveillance video is not sufficient. Therefore, from a practical perspective, this paper proposes a fast sparse 3D convolution operation and forms a corresponding 3D-SCNN to test and demonstrate the data set.

3. System architecture

We have developed a video monitoring and analysis platform using the SIP protocol and GB/T28181. The system structure is shown in Fig. 1. The platform consists of a coordination server, access and signaling server, storage server, codec server, video analysis server, monitoring terminal, etc. It can realize the monitoring resources, sharing, storage and distribution of the entire network, and can be expanded and upgraded to have application functions such as intelligent monitoring, email SMS alarm and mobile phone monitoring.

As shown in Fig. 1, the coordinating server implements the business management functions of device management, user management, authority management and log management; the signaling/access server is responsible for the access of all terminal devices, which is specifically reflected in the registration and positioning services, heartbeat services, message forwarding services, redirection services and proxy services of the front-end devices. The main function of the storage server is to store historical recordings and quickly retrieve them; the monitoring terminal is the core part of the system, which connects various monitoring resources, decodes and stores the required video information, and plays, retrieves and browses in real time according to authorization, obtains video analysis results and alarm information, and uses the message service implemented by Web Services to provide decision-making and reliable monitoring information for departments and users at all levels.

The platform prioritizes compliance with the GB/T28181 standard. For video resources that meet the GB/T28181 standard, access the video surveillance platform through the video resources; for video resources or platforms that do not meet the GB/T28181 standard, access the video resource integration platform through the access gateway method (protocol conversion), and finally access the video surveillance platform. If the platform is accessed through the access gateway for protocol conversion, the front-end device must support the ONVIF protocol [8] or provide an SDK package, access the protocol conversion server, and then access the video surveillance platform through the protocol conversion server.

4. Proposed fall detection approach based on 3D-SCNN

A typical CNN network usually alternately stacks multiple convolutional layers and pooling layers to process and compress the input signal, and finally completes the mapping between features/targets through a fully connected layer. In 2D CNN, convolution and pooling operations only reflect the spatial dimension, which is not very effective for processing video streams that contain both spatial and temporal information.

4.1. 3D Convolutions

In the convolution layers of CNN, convolution is a special linear operation between input data and multiple convolution kernel functions to generate convolution feature maps. 3D convolution preserves and abstracts temporal and spatial information by convolving the 3D convolution kernel with a video cube consisting of multiple adjacent frames.

In general, 2D convolution is to slide the 2D convolution kernel in the spatial dimension for convolution operation; while 3D convolution is to slide the 3D convolution kernel in both spatial and temporal dimensions for convolution operation. In video analysis, each position of the 3D feature map is connected to multiple adjacent input frames, thereby retaining certain temporal information of the input frames.

Similar to the convolution operation in the two-dimensional case, assuming that the input tensor of the convolution layer L in the three-dimensional case is $x^L \in$

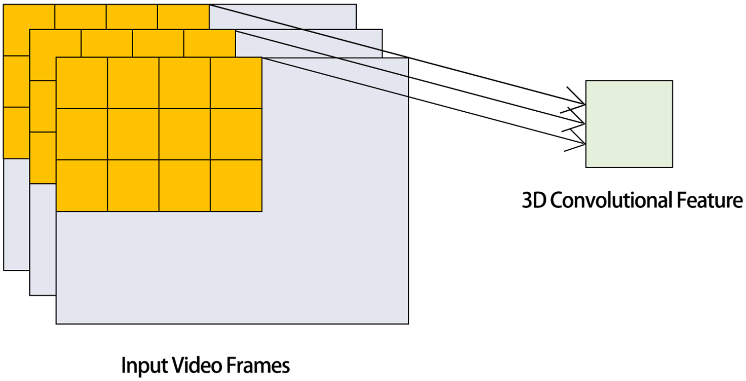


Fig. 2. 3D Convolution kernel and convolution features.

$\mathbb{R}^{H^L \times W^L \times D^L}$, the convolution kernel of this layer is $f^L \in \mathbb{R}^{H \times W \times D^L}$. The 3D convolution is actually to expand the two-dimensional convolution to all channels of the corresponding position (i.e., D^L), and finally sum up all HWD^L elements processed by one convolution as the convolution result of this position.

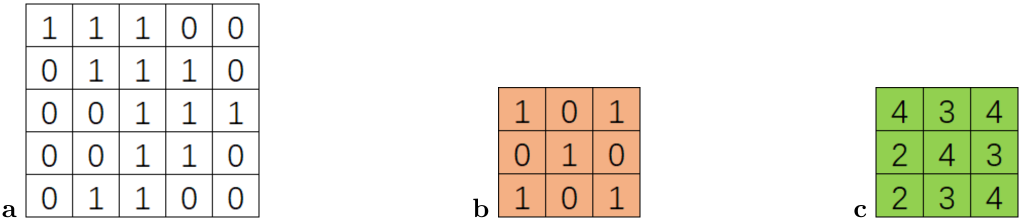
As shown in the Fig. 2, the convolution kernel size is $3 \times 4 \times 3$, and the output result of $1 \times 1 \times 1$ is obtained after convolution at this position. Furthermore, if there are D convolutions like f, the convolution output of $1 \times 1 \times 1 \times D$ dimensions can be obtained at the same position, and D^{L+1} is the number of channels of the $L + 1$-th layer feature x^{L+1}.

4.2. 3D Sparse Convolutions

In the convolution layers of CNN, convolution is a special linear operation between input data and multiple convolution kernel functions to generate convolution feature maps. 3D convolution preserves and abstracts temporal and spatial information by convolving the 3D convolution kernel with a video cube consisting of multiple adjacent frames.

In view of the slow convolution speed of traditional 3D-CNN, we have proposed a sparse convolution method 3D Sparse Convolutions and the corresponding 3D Sparse Convolutional Neural Network (3D-SCNN), which can achieve faster convolution at the approximate accuracy, thereby improving the operation speed of the entire convolution network. Now take the convolution operation in two-dimensional mode as an example:

As shown in Fig. 3, the traditional convolution operation uses a 3×3 convolution kernel (as shown in Fig. 3b), and directly performs sliding convolution on the 5×5 element matrix (as shown in Fig. 3a). Each time it slides, a convolution value is obtained. It takes 9 convolution operations to get the final convolution matrix (as shown in Fig. 3c).



a

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

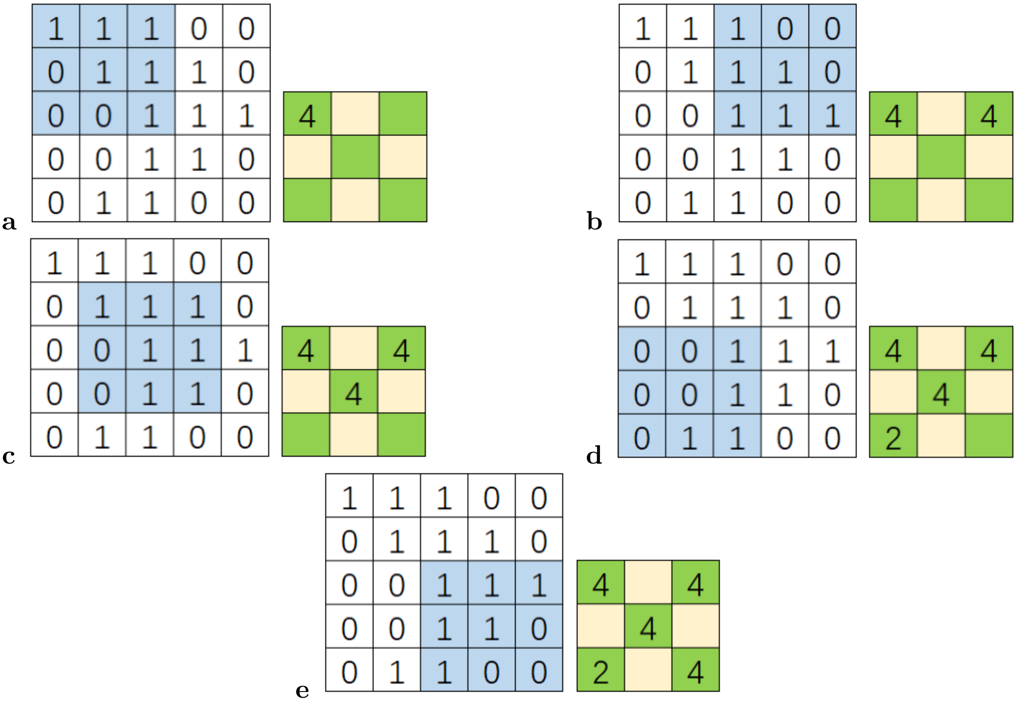
b

1	0	1
0	1	0
1	0	1

c

4	3	4
2	4	3
2	3	4

Fig. 3. The original convolution operation. (**a**) Element matrix; (**b**) convolution kernel; (**c**) convolution matrix.



a

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		

b

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		4

c

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		4
	4	

d

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		4
	4	
2		

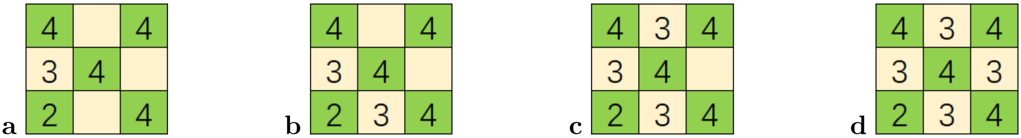
e

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

4		4
	4	
2		4

Fig. 4. Five convolution operations in sparse convolution. (**a**) Operation one; (**b**) Operation two; (**c**) Operation three; (**d**) Operation four; (**e**) Operation five.

However, using the Sparse Convolution Method we proposed only requires 5 convolution operations and 4 simple algebraic operations with low computational complexity. As shown in Fig. 4, the specific steps of sparse convolution are: using a 3×3 convolution template, in a 5×5 image, convolution is performed once every sliding window, so that the numerical scale after the actual convolution is only half of the scale of the ordinary



a

4		4
3	4	
2		4

b

4		4
3	4	
2	3	4

c

4	3	4
3	4	
2	3	4

d

4	3	4
3	4	3
2	3	4

Fig. 5. Four simple algebraic operations in sparse convolution. (**a**) Operation one; (**b**) Operation two; (**c**) Operation three; (**d**) Operation four.

convolution, and then for the skipped sliding window, in the final convolution matrix, first find the smallest convolution result unit, and then calculate the average value of it and its direct convolution result unit,

$$c_{i,j} = \begin{cases} \mathrm{Avg}(c_{i-1,j}, c_{i+1,j}) \\ \mathrm{Avg}(c_{i,j-1}, c_{i,j+1}) \\ \mathrm{Min}(\mathrm{Adj}(c_{i,j})) \end{cases} \tag{1}$$

for other space convolution result units, use the convolution value of the smallest neighbor as its value,

$$\mathrm{Adj}(c_{i,j}) = \{c_{i-1,j-1}, c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j+1}, c_{i+1,j-1}, c_{i+1,j}, c_{i+1,j+1}\} \tag{2}$$

As shown in Fig. 5, then for the skipped sliding window, in the final convolution matrix, first find the smallest convolution result unit 2, that is, $c_{3,1}$, and then calculate the average of $c_{3,1}$ and its two directly adjacent element points $c_{1,1}$ and $c_{3,3}$, and get 3 and 3, which are used as the convolution value 3 of $c_{2,1}$ and the convolution value 3 of $c_{3,2}$ respectively. For the other space convolution result units, $c_{1,2}$ and $c_{2,3}$, use the smallest neighboring convolution value 3 as their value. This convolution method is not limited to 3 × 3 convolution, and its deformation is also applicable to void convolution.

The three-dimensional case is relatively easy to expand. We only need to replace the Adj function with a three-dimensional version, that is, select appropriate adjacent elements in the three-dimensional space according to a certain rule. The elements whose spatial distance from it is 1 are relatively easy to achieve.

4.3. 3D Sparse Convolutional Neural Network: 3D-SCNN

The overall system architecture usually includes multiple components that work together to achieve the desired functionality. Such network architectures typically include multiple layers of 3D convolutions, pooling, normalization, and possibly recurrent layers or attention mechanisms to improve performance. The network is trained using labeled data and the model learns to recognize patterns and features associated with different actions or events in the video sequence. Optimization techniques such as gradient descent and regularization methods are applied to improve the accuracy and generalization of the model. After the network has processed the video sequence, post-processing steps may involve temporal pooling, which aggregates features in time to make a final decision about the presence or absence of a specific action or event. Thresholding or classification algorithms can be used to interpret the output of the network and make predictions. After validation, the system can be deployed to real-world applications. This may involve integration with existing software or hardware systems to ensure compatibility and performance in the operating environment.

The network input has been reshaped into a set of 36 depth images, each of size 64x64 pixels. The length of the video segments has been experimentally set to achieve a compromise between processing time and action recognition accuracy. Since the image size in the dataset is 640 × 480 pixels, a preprocessing stage is required to crop the original depth input images into square images and then resize them to the required dimensions.

The designed algorithm therefore uses a sliding window to select the frames to be inserted into the network each time. The window size is 36 frames and the stride is 12 frames, so that there is a dynamic number of frames overlap between consecutive input sets. Each 36-frame segment is then processed to obtain a final vector that includes its classification probability of belonging to one of the five possible actions.

In the proposed network (see Fig. 6), the first two layers ("Sparse Conv3D 1", "Sparse Conv3D 2" and "Sparse Conv3D 3") are convolutional layers with 32 filters each and kernel size (3,3,3). To avoid dimensionality reduction in these layers, Sparse Conv3D operations are used. This is followed by a pooling layer ("Max Pooling 1") for dimensionality reduction. The resulting output tensor is then fed into another pair of convolutional layers "Sparse Conv3D 3" and "Sparse Conv3D 4", each with 64 filters and kernel size (3,3,3) to extract features with a higher level of abstraction.

Several "Sparse Conv 3D" layers use ReLU (Rectified Linear Unit) activation functions on their outputs. This type of activation function provides the necessary nonlinearity for the classifier for action detection. The "Dropout" technique is also used during training to randomly ignore nodes in the network at each training stage to prevent overfitting.

4.4. Preprocessing

Preprocessing techniques include parsing the video stream, selecting key frames, applying background subtraction, resizing frames, converting frames from RGB to grayscale mode, and finally using the processed frames as input to the 3D-SCNN model. The data is then normalized to scale the pixel values to between 0 and 1. In addition to this, data labeling is performed to assign corresponding labels to the training data and validation data based on the number of positive and negative samples read, and the labels are one-hot encoded for comparison with the model output. Since the image size in the dataset is 640 × 480 pixels, a preprocessing stage is required to crop the original depth

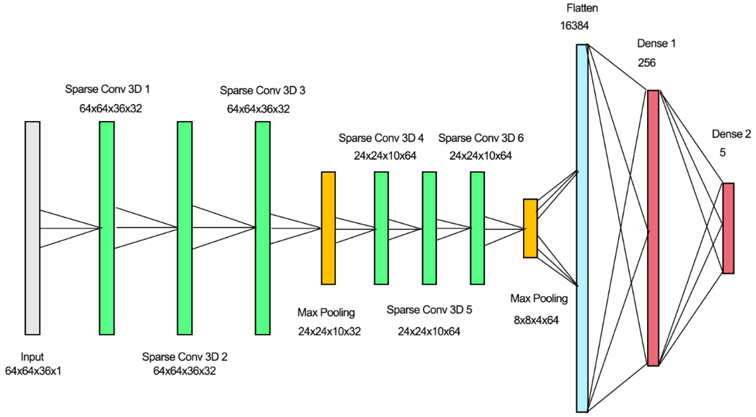


Fig. 6. 3D Sparse Convolutional Neural Network: 3D-SCNN.

input image into a square image and then resize it to the required size. Each image is first cropped into a square of 240 × 240 pixels and then resized to the network input size (64 × 64 pixels).

4.4.1. Key frames selection

Generally speaking, the movements of the elderly are not as intense as those in action movies, and considering the processing time and delay, we also optimized the selection of key frames.

In the system, we introduced the Jitter buffer [7] in WebRTC [28] for management, which can not only help deal with delay and jitter in the video stream, but also affect the selection of key frames. There are usually three ways to use the jitter buffer to select key frames.

a) Timing selection method: this method selects key frames from the jitter buffer based on a fixed time interval. For example, a key frame is selected from the buffer at a certain interval. This method is simple and direct, suitable for applications with low real-time requirements, so it cannot be used directly in our system.

b) Threshold selection method: Set a threshold, and select a key frame when the amount of data in the jitter buffer exceeds or falls below this threshold. For example, when the amount of data in the jitter buffer is low, select a key frame to fill the buffer to prevent data loss or maintain a smooth flow of data. However, how to choose a suitable threshold is also not easy to determine in automated video analysis.

c) Dynamic adjustment method: This method dynamically adjusts the key frame selection strategy according to the current network conditions and the state of the buffer.

For example, when the network jitter is large, key frames can be selected first to ensure the continuity and quality of the video stream.

The dynamic adjustment method is mainly to dynamically adjust the key frame selection strategy based on the current network status and the status of the buffer when selecting key frames using the jitter buffer. The goal of this method is to effectively use the buffer to handle network jitter and delay while ensuring video quality and continuity. The specific operation may include the following steps: First, the network delay, jitter and data volume in the jitter buffer must be monitored in real time. This can be achieved by monitoring the water level in the jitter buffer and the length of the decoding buffer. Second, set the adjustment strategy: According to the monitored network conditions and buffer status, set the adjustment strategy for key frame selection. For example: When the network jitter is small and there is enough data in the jitter buffer, fewer key frames can be selected (that is, the time interval for selecting key frames can be increased) to reduce bandwidth consumption and improve playback efficiency. When the network jitter is large or the amount of data in the jitter buffer is insufficient, all key frames can be selected first to ensure the continuity and quality of the video stream.

In order to ensure the continuity of features, we deliberately overlapped the cubes composed of multiple frames each time, which ensured the integrity and continuity of convolutional features. The network input sequence must be divided into 36-frame segments and then analyzed by 3D-CNN. The designed algorithm therefore uses a sliding window to select the frames to be inserted into the network each time. The window size is 36 frames and the initial stride is 12 frames, so that there is an overlap of 12 frames between consecutive input sets. In the study, we made the jitter buffer also affect the size of the overlap stride. When the network jitter is small and there is enough data in the jitter buffer, the overlap stride gradually decreases from the initial value of 12 to a minimum of 4; when the network jitter is large or the amount of data in the jitter buffer is insufficient, the initial value of 12 gradually increases to a maximum of 18. Subsequent experiments show that dynamic overlap control based on sliding windows will enhance the robustness of fall detection.

4.4.2. Background segmentation

In the preprocessing stage, we algorithmically distinguish the moving area and background from the video and classify the foreground as human or non-human. Among the many background removal algorithms, we choose ViBE [5] which is friendly to video surveillance analysis and can also adjust the background model online. It is divided into the following steps: initialization, foreground detection and model update. The first frame is initialized as a background frame with 20 background samples per pixel. Then, each pixel is labeled as foreground or background based on its historical value. The last step is to select updateable neighboring pixels and randomly determine them as background pixels. Background samples are randomly selected to update the model, while

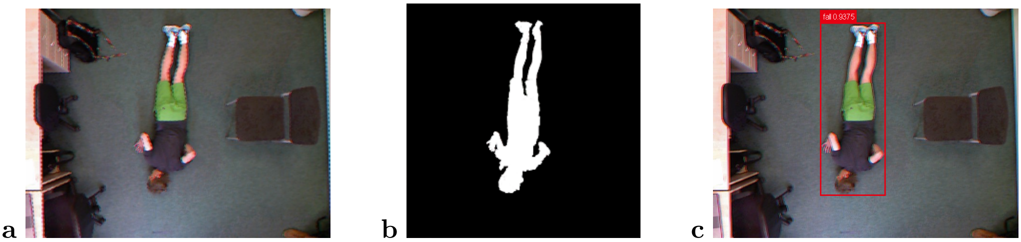


Fig. 7. Background segmentation in the preprocessing. (**a**) Original frame image; (**b**) reduced square segmented image; (**c**) final frame image with synthesized recognition box.

other samples are discarded. To ensure that the foreground region contains only humans, we further refine the foreground segmentation through morphological operations to eliminate the most invalid regions. Then, the connected components are calculated and if their size is smaller than the minimum human size, they are removed from the foreground; otherwise, they are considered as human regions. Therefore, a foreground region containing only human objects is obtained. Finally, the detection is verified based on the percentage of foreground pixels within the bounding box of each detected person. If this value is lower than the threshold, the detection result is considered a false positive.

As shown in Fig. 7, after preprocessing, the original frame image shown in Fig. 7a is obtained as a reduced square segmented image Fig. 7b. A series of segmented images are used as the input of 3D-SCNN. After forward propagation of the network, the result is finally re-enlarged and composed with the original frame image with a synthesized recognition box Fig. 7c.

In the experiment, we build a 3D-SCNN model, added multiple Conv3D layers, MaxPooling3D layers, Flatten layers, and fully connected layers (Dense layers), and added Dropout layers to prevent overfitting. Finally, compile the model, specify the loss function as categorical cross entropy, the optimizer as SGD, and specify the metrics as accuracy and mean square error (MSE). Then train the model and pass in the training data, validation data, and callback function (for saving the best model). Save the trained model file after training.

The pooling layer is set after the convolution layer, and the feature image output by the convolution layer is pooled, also known as down sampling, which can reduce the dimension of the feature and improve the generalization ability of the model. There are two classic pooling methods – average pooling and maximum pooling. In addition to the convolution layer, the pooling layer is also a major component of a typical CNN. It subsamples the feature map transmitted from the convolution layer according to the principle of local correlation. The pooling operation outputs the summary statistics of the neighboring units at a certain position in the feature map, thereby reducing the amount of data while retaining valuable information. Similarly, we apply 3D maximum pooling to achieve translation invariance of cubic video patches in spatial and temporal dimensions.

5. Experiments

5.1. Experimental setup

The experiment machine was an Intel® Core™ i3-1220P processor running at 4.40 GHz, 10 Cores, 12 MB Intel® Smart Cache, 64 GB of RAM, two NVIDIA GeForce RTX 2080Ti graphics processing units, and a 64-bit Windows 10 operating system.

For training, we used two dataset splits: 85% and 95% of the data for training, 10% and 3% of the data for validation, and finally 5% and 2% of the data for testing. Through 1800 rounds of training, the proposed 3D-SCNN method achieved good accuracy under different input conditions. The stochastic gradient descent optimizer and cross entropy loss function were used. The results were refined by dropout of 0.2, and the learning rate was set to 0.001.

5.2. Datasets

In the test and validation phase, we use the following two datasets:

1. **Multi-camera** [4] In this dataset, the authors collected data of falls and normal activities from a calibrated multi-camera system consisting of 8 inexpensive IP wide-angle cameras that can cover the entire room. The 8 cameras captured 22 fall scenes, including sequences of falling forward or backward while walking, falling when sitting in an improper posture, losing balance, etc., as well as 2 normal daily activity scenes, such as walking in different directions, doing housework, and activities with similar characteristics to falls (sitting/standing up, squatting).
2. **UR Fall dataset** [14] In this dataset, the authors collected data containing 70 videos, including 30 fall videos and 40 daily life activity videos. Experiments were conducted on the UR fall dataset, which contains 30 fall and 40 non-fall depth videos, obtained from top view and front view, respectively. The fall action categories include falling while walking, sudden falling, and falling from a chair. The non-falling action categories include walking, sitting, bending over and other scenes.

The performance of the system is measured by its accuracy on the test data. Accuracy is calculated by comparing the predicted labels for each set of 36 frames with the labels assigned to the entire sequence. As explained before, each video sequence is processed using a sliding window and one result (label) is obtained per window. The system then assigns the label to the action with the highest calculated probability.

In Table 1, we report the performance of various deep neural network algorithms on the fall detection dataset. Since fall detection is actually a binary classification problem,

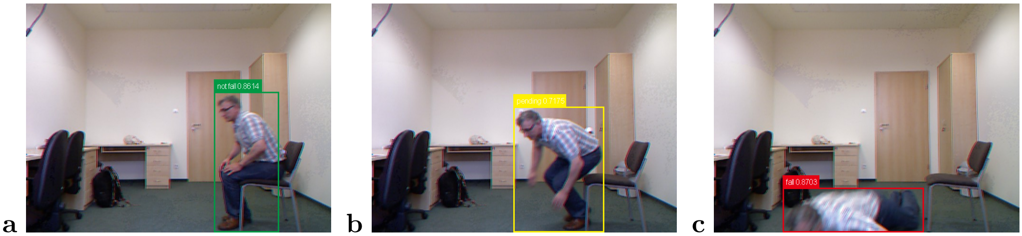


Fig. 8. Fall detection with 3D-SCNN. (**a**) Not fall; (**b**) pending; (**c**) fall.

we report the accuracy of 96.59% and 99.82% for the 3D-SCNN. We find that our 3D-SCNN architecture outperforms the 3D-CNN for fall detection on the UR Fall dataset, and is slightly inferior to the 3D-CNN for multi-camera fall detection. The overall accuracy is stronger than the VAG [18], 3D-CNN [18], OpenPose, LSTM/GRU [19], YOLO, OpenPose, Random Forest [19], and FPD [25] tested in our environment which is slightly low than the data in the original papers. In addition, compared with other methods, such as Kraft et al. [17] and Chahyati et al. [9], our accuracy is higher than theirs. However, compared with Umar Asif [3], our accuracy is slightly lower, which may be related to the learning of human skeletons and segmentation-based fall representations from synthetic data in the article. These modeling may lead to improved accuracy. In all fall sequences, the fall action is correctly detected in multiple 36-frame windows. In addition, the system can accurately predict other actions of interest, such as walking or running, and the accuracy of both actions exceeds the applicable level.

The model divides the elderly's behaviors into three categories: falling, not falling, and pending. As shown in the Fig. 8, after being tested on the UR Fall dataset, the system can basically accurately identify these three behaviors. However, since pending

Tab. 1. Performance of various approaches on Multi-camera fall dataset and UR fall dataset

Method	Multi-Camera Dataset	UR Dataset
FPD [25]	96.20%	-
3D-CNN [18]	99.73%	-
VAG 3D-CNN] [18]	99.367%	-
OpenPose, LSTM/GRU [19]	-	98.2 %
YOLO, OpenPose [19]	-	97.33%
Kraft et al. [17]	-	95.2 %
Chahyati et al. [9]	-	95.64%
Umar Asif [3]	98.60%	-
Proposed 3D-SCNN	96.59%	99.82%

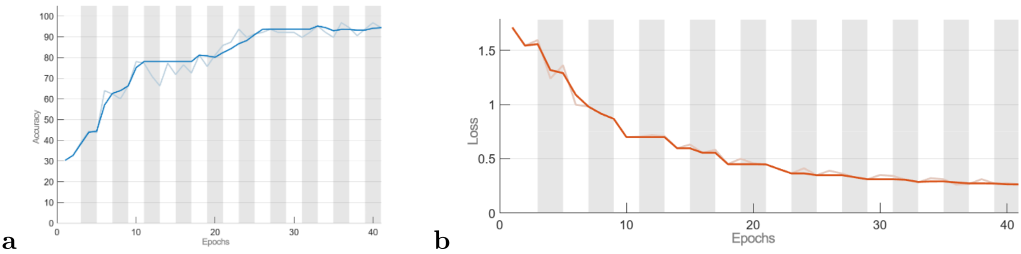


Fig. 9. Traditional 3D Convolutional Neural Network. (**a**) Accuracy of training and validation; (**b**) losses of training and validation.

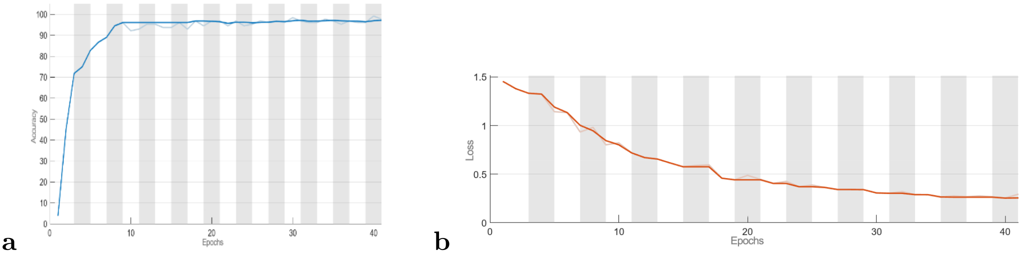


Fig. 10. Sparse 3D Convolutional Neural Network. (**a**) Accuracy of training and validation; (**b**) losses of training and validation.

behavior actually includes a variety of abnormal behaviors, the model is slightly lacking in the recognition of pending. The system will give a certain warning to the caregiver when pending, confirm the alarm notification when falling, and save the relevant video for medical caregivers to diagnose and analyze. After testing on the UR Fall dataset, we found that the model can basically meet the accuracy requirements of fall detection in home scenarios after training.

Fig. 9 shows the accuracy and loss graph of the 3D-CNN model with the traditional convolution proposed previously. We obtained a large number of parameters for the fully connected layers and completed the learning within 100 epochs. Fig. 10 shows the accuracy and loss graph of the 3D-SCNN model with the addition of sparse convolution. The accuracy of this model shows a significant improvement, and the training is completed at the 82nd epoch. It not only reduces the number of parameters, but also reduces the training time per epoch and the number of convolution operations during testing, and obtains an accuracy of more than 96%.

6. Discussion

6.1. Differences and advantages between 3D-SCNN and traditional 3D-CNN

Here, we mainly discuss the differences and advantages of 3D-SCNN over traditional 3D-CNN. Traditional 3D-CNN perform standard convolution operations on the entire input data. Each convolution layer applies a fixed-size convolution kernel to perform convolution operations on the entire 3D input space to process all the information in the input data. Due to full convolution calculations, traditional 3D-CNNs usually have high memory usage, especially when processing large-scale 3D data (such as video frames or depth maps). In hardware-constrained environments, training and inference speeds may be slow, especially for tasks involving a large number of frames or long time series. 3D-SCNN (sparse convolutional neural network) uses sparse convolution, which can capture useful features more efficiently and avoid unnecessary calculations. For the same size of input data, 3D-SCNN is usually much faster to train than traditional 3D-CNN, and can also save computing resources during inference. It is particularly suitable for applications such as video surveillance that require real-time processing, and can also leave a lot of valuable time for later analysis and early warning operations.

6.2. Limitations of 3D-SCNN

Although 3D-SCNN exhibits significant advantages over traditional 3D-CNN in terms of computational efficiency, memory footprint, and performance, it also suffers from some limitations. The following are the main limitations of 3D-SCNN:

(1) 3D-SCNN performs well in sparse data (for example, most areas are empty or background), but when the input data is very dense or contains a lot of information, the advantages of sparse convolution may be weakened. In this case, traditional 3D-CNN may be more suitable as it can process the entire input space uniformly without relying on the sparsity of the data. When faced with applications where every region in the data contains important information (such as fine-grained action recognition or fine image analysis), the advantages of sparse convolution may not be so obvious, and may even lead to feature loss, affecting the performance of the model.

(2) Sparse convolution requires specially designed hardware or software support to efficiently handle sparse data. This makes the implementation of the model more complex than traditional 3D-CNN, especially in the absence of specialized optimization libraries or hardware acceleration, and the efficiency improvement of sparse convolution may not be fully reflected. For systems without hardware acceleration (such as GPU or custom hardware), the implementation of sparse convolution may be more time-consuming than standard convolution, and may even reduce computational efficiency and increase development costs in some applications.

7. Conclusions

In this paper, a fall detection system based on 3D sparse convolutional neural network (3D-SCNN) is proposed, which uses new sparse convolution to replace the traditional convolution operation, extracting features in spatial and temporal dimensions from depth information to predict the fall action in the scene. Additionally, the preprocessing stage involves a dynamic key frame selection approach, using the jitter buffers to adjust frame selection based on current network conditions and buffer state. To ensure feature continuity, overlapping cubes of selected frames are intentionally employed, with dynamic resizing to adapt to network dynamics and buffer states. Experiments are conducted on Multi-camera fall dataset and UR fall dataset, and the results show that its accuracy exceeds the three compared methods, and outperforms the traditional 3D-CNN method in both precision and loss.

Despite the effectiveness of the system in detecting human falls from video, the proposed method has several limitations. Firstly, it can only detect single-person falls without specific localization of individuals in the scene. Secondly, it cannot accurately detect falls involving complex movements, such as when an elderly person falls from a sitting to standing position. Additionally, methods based on 3D-SCNNs suffer from high computational demands, resulting in delays during practical applications.

In the future research, we will enhance the performance of the improved 3D-SCNN method across various publicly available datasets to pump its generalizability and effectiveness in real-world environment. Also, we aim to investigate conditions involving multiple cameras to provide faster and more accurate detection and alerts. Furthermore, we will explore integration with wearable sensor-based methods to offer more intelligent solutions.

Conflicts of interest

The author declare no conflict of interest.

References

[1] T. Alanazi, K. Babutain, and M. Ghulam. Mitigating human fall injuries: A novel system utilizing 3D 4-stream convolutional neural networks and image fusion. *Image and Vision Computing* 148:105153. 2024. doi:10.1016/j.imavis.2024.105153.

[2] T. Alanazi and G. Muhammad. Human fall detection using 3D multi-stream convolutional neural networks with fusion. *Diagnostics* 12(12):3060. 2022. doi:10.3390/diagnostics12123060.

[3] U. Asif, B. S. Mashford, S. von Cavallar, S. A. C. Yohanandan, S. Roy, et al. Privacy preserving human fall detection using video data. In: *Proceedings of the Machine Learning for Health NeurIPS Workshop*, vol. 116 of *Proceedings of Machine Learning Research*, pp. 39–51. 2020. `https://proceedings.mlr.press/v116/asif20a/asif20a.html`.

[4] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. *Multiple cameras fall dataset.*

Technical Report 1350, DIRO-Université de Montréal. Jul 2010. `https://www.iro.umontreal.ca/~labimage/Dataset/`.

[5] O. Barnich and M. V. Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing* 20(6):1709–1724. 2011. doi:10.1109/TIP.2010.2101613.

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733. 2017. doi:10.48550/arXiv.1705.07750.

[7] Y. Cinar, P. Pocta, D. Chambers, and H. Melvin. Improved jitter buffer management for WebRTC. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17(1):30. 2021. doi:10.1145/3410449.

[8] H. W. Di, C. Y. Luo, and X. C. Cai. Research and application of ONVIF protocol in IP camera. In: *Measurement Technology and its Application III*, vol. 568 of *Applied Mechanics and Materials*, pp. 1399–1402. Trans Tech Publications Ltd. 2014. doi:10.4028/www.scientific.net/AMM.568-570.1399.

[9] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, et al. Application of convolutional neural networks for fall detection using multiple cameras. In: *Challenges and Trends in Multimodal Fall Detection for Healthcare*, pp. 97–120. Springer International Publishing, Cham. 2020. doi:10.1007/978-3-030-38748-8_5.

[10] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6201–6210. 2019. doi:10.1109/ICCV.2019.00630.

[11] L. B. Freire, J. P. Brasilneto, L. D. S. Marianne, M. G. Cruz Miranda, et al. Risk factors for falls in older adults with diabetes mellitus: systematic review and meta-analysis. *BMC Geriatrics* 24(1):201. 2024. doi:10.1186/s12877-024-04668-0.

[12] R. Jayaswal, A. Pathak, and S. Mahajan. Integrating 3dcnn attention mechanism with pose estimation for indoor fall detection. Available at SSRN, preprint 4883239. doi:10.2139/ssrn.4883239.

[13] S. Jiang, N. Liu, and G. Yang. Design and implementation of WebRTC video conference system structure compatible with GB/T28181 devices*. In: *Proceedings of the 2022 International Conference on Computer Science, Information Engineering and Digital Economy (CSIEDE 2022)*, pp. 411–420. Atlantis Press. 2022. doi:10.2991/978-94-6463-108-1_47.

[14] M. Kepski and B. Kwolek. Fall detection on embedded platform using Kinect and wireless accelerometer. In: *Computers Helping People with Special Needs (ICCHP 2012)*, vol. 7383 of *Lecture Notes in Computer Science*, pp. 407–414. 2012. doi:10.1007/978-3-642-31534-3_60.

[15] A. I. Khan, S. Jain, and P. Sharma. A new approach for human identification using ai. *2022 International Mobile and Embedded Technology Conference (MECON)* pp. 645–651. 2022. doi:10.1109/MECON53876.2022.9752153.

[16] F. Kinoshita and H. Takada. Numerical analysis of stochastic differential equations describing body sway while viewing 3D video clips. *Mechatronic Systems and Control* 47(2):98–105. 2019. doi:10.2316/J.2019.201-2995.

[17] D. Kraft, K. Srinivasan, and G. Bieber. Deep learning based fall detection algorithms for embedded systems, smartwatches, and IoT devices using accelerometers. *Technologies* 8(4):72. 2020. doi:10.3390/technologies8040072.

[18] N. Lu, Y. Wu, L. Feng, and J. Song. Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data. *IEEE Journal of Biomedical and Health Informatics* 23(1):314–323. 2019. doi:10.1109/JBHI.2018.2808281.

[19] N. Mamchur, N. Shakhovska, and M. Gregus ml. Person fall detection system based on video stream analysis. *Procedia Computer Science* 198:676–681. 2022. doi:10.1016/j.procs.2021.12.305. 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare.

[20] National Academies of Sciences, Engineering, and Medicine. *Social Isolation and Loneliness in Older Adults: Opportunities for the Health Care System*. The National Academies Press, Washington, DC. 2020. doi:10.17226/25663.

[21] J. Nogas, S. S. Khan, and A. Mihailidis. DeepFall: Non-invasive fall detection with deep spatio-temporal convolutional autoencoders. *Journal of Healthcare Informatics Research* 4:50–70. 2019. doi:10.1007/s41666-019-00061-4.

[22] N. Noor and I. K. Park. A lightweight skeleton-based 3D-CNN for real-time fall detection and action recognition. In: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2179–2188. 2023. doi:10.1109/ICCVW60793.2023.00232.

[23] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras. Vision-based fall detection with convolutional neural networks. *Wireless Communications and Mobile Computing* 2017(1):9474806. 2017. doi:10.1155/2017/9474806.

[24] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5534–5542. 2017. doi:10.1109/ICCV.2017.590.

[25] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on Circuits and Systems for Video Technology* 21(5):611–622. 2011. doi:10.1109/TCSVT.2011.2129370.

[26] E. Schooler, J. Rosenberg, H. Schulzrinne, A. Johnston, G. Camarillo, et al. SIP: Session Initiation Protocol. In: *RFC*, no. 3261 in Request for Comments. RFC Editor. Jul 2002. doi:10.17487/RFC3261.

[27] C. Silver and T. Akilan. A novel approach for fall detection using thermal imaging and a stacking ensemble of autoencoder and 3D-CNN models. In: *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 71–76. 2023. doi:10.1109/CCECE58730.2023.10288941.

[28] B. Sredojev, D. Samardzija, and D. Posarac. WebRTC technology overview and signaling solution design and implementation. In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1006–1009. 2015. doi:10.1109/MIPRO.2015.7160422.

[29] C. Su, J. Wei, D. Lin, L. Kong, and Y. L. Guan. A novel model for fall detection and action recognition combined lightweight 3D-CNN and convolutional LSTM networks. *Pattern Analysis and Applications* 27(1):3. 2024. doi:10.1007/s10044-024-01224-9.

[30] M. M. Sylaja and J. Kurian. Robot task recognition using deep convolutional long short-term memory. *Mechatronic Systems and Control* 51(2):106–113. 2023. doi:10.2316/J.2023.201-0353.

[31] A. Sánchez-Caballero, S. de López-Diz, D. Fuentes-Jimenez, C. Losada-Gutiérrez, M. Marrón-Romera, et al. 3DFCNN: real-time action recognition using 3D deep neural networks with raw depth information. *Multimedia Tools and Applications* 81(17):24119–24143. 2022. doi:10.1007/s11042-022-12091-z.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497. 2015. doi:10.1109/ICCV.2015.510.

[33] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, et al. A closer look at spatiotemporal convolutions for action recognition. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459. 2018. doi:10.1109/CVPR.2018.00675.

[34] X. Xiong, W. Min, W. Zheng, P. Liao, H. Yang, et al. S3D-CNN: skeleton-based 3D consecutive-low-pooling neural network for fall detection. *Applied Intelligence* 50:3521–3534. 2020. doi:10.1007/s10489-020-01751-y.

[35] C. Xu. Extracting and recognising music features through multi-modal emotion recognition. *Mechatronic Systems and Control* 52(3):140–146. 2024. doi:10.2316/j.2024.201-0380.

[36] J. Zou and H. Zhang. New key point detection technology under real-time eye tracking. *Mechatronic Systems and Control* 47(2):71–76. 2019. doi:10.2316/J.2019.201-2969.

Fangping Fu received her Master's degree in Social Work from the Sun Yat-sen University, Guangzhou, China, in 2010. Since 2011 she is with the Shunde Polytechnic, Foshan, China. Her research interests comprise Social Work and Elderly Care Services. Previously, she served as an evaluation expert for the Guangzhou Social Work Association and the Shunde District Social Work Federation.