A Hybrid Multi-Person Fall Detection Scheme Based on Optimized YOLO and ST-GCN

Lei Liu[1], Yeguo Sun[2]*, Xianlei Ge[3]

[1] School of Computer Science, Huainan Normal University, Huainan (China)
[2] School of Finance and Mathematics, Huainan Normal University, Huainan (China)
[3] School of Electronic Engineering, Huainan Normal University, Huainan (China)

* Corresponding author: yeguosun@126.com

Received 21 July 2023 | Accepted 3 June 2024 | Early Access 26 September 2024

Abstract

Human falls are a serious health issue for elderly and disabled people living alone. Studies have shown that if fallers could be helped immediately after a fall, it would greatly reduce their risk of death and the percentage of them requiring long-term treatment. As a real-time automatic fall detection solution, vision-based human fall detection technology has received extensive attention from researchers. In this paper, a hybrid model based on YOLO and ST-GCN is proposed for multi-person fall detection application scenarios. The solution uses the ST-GCN model based on a graph convolutional network to detect the fall action, and enhances the model with YOLO for accurate and fast recognition of multi-person targets. Meanwhile, our scheme accelerates the model through optimization methods to meet the model's demand for lightweight and real-time performance. Finally, we conducted performance tests on the designed prototype system and using both publicly available single-person datasets and our own multi-person dataset. The experimental results show that under better environmental conditions, our model possesses high detection accuracy compared to state-of-the-art schemes, while it significantly outperforms other models in terms of inference speed. Therefore, this hybrid model based on YOLO and ST-GCN, as a preliminary attempt, provides a new solution idea for multi-person fall detection for the elderly.

Keywords

Computer Vision, Elderly Protection, Fall Detection, Graph Convolutional Network, Human Pose Estimation.

DOI: 10.9781/ijimai.2024.09.003

I. Introduction

The elderly population is growing faster than any other age group, and as of October 2022, 10% of the world's total population will be over 65 years old [1]. Related studies predict that the total number of older adults will increase to 1.5 billion by the end of 2050 [2]. Older people's physical, cognitive, and motor skills decline with age. Falls are a significant challenge for them, and they can significantly reduce the life expectancy of older adults. Approximately 35% of people (65 years and older) fall once or more yearly [3]. In addition to old age, other factors such as environment, physical action, and cardiovascular disease can contribute to falls. It is a significant source of physical injuries, and these injuries usually require hospitalization for long-term treatment [4]. Each year, 37.3 million falls require medical care and 650,000 falls result in death [5]. In Fig.1, medical investigations have shown that timely treatment after a fall can reduce the risk of death by 80% and significantly improve the survival rate of older adults. Therefore, rapid detection of fall events is of great significance [6]. Accurate human action recognition methods and model optimization techniques are vital to achieving this goal. However, there are also problems, such as significant differences in the structure and performance of each different model, low support for multi-person action recognition, and low real-time system performance [7]. This paper proposes a hybrid human fall detection framework based on YOLO and ST-GCN for multiple people for the elderly fall detection scenario in real time. Two optimization algorithms accelerate the model to improve real-time performance and accuracy of the model.

Fig. 1. Elder fall and timely treatment.

Please cite this article as:
L. Liu, Y. Sun, X. Ge. A Hybrid Multi-Person Fall Detection Scheme Based on Optimized YOLO and ST-GCN, International Journal of Interactive Multimedia and Artificial Intelligence, (2024), http://dx.doi.org/10.9781/ijimai.2024.09.003

The primary advantages of the scheme are summarized as follows:

- We attempt to recognize human actions using skeletons and propose a hybrid model based on YOLO and ST-GCN to improve adaptation to multi-person fall detection scenarios.
- We accelerate the proposed hybrid model by using two model optimization algorithms to reduce the model size and improve the scheme's overall real-time performance.
- We construct our own fall detection test dataset for multi-person fall detection, and analyze the causes of miss and false detection in the fall detection model and provides references for subsequent research.

The rest of this paper is organized as follows: Section 2 reviews the research related to human fall detection and model optimization. Section 3 presents the proposed scheme's overall design framework and each component's functions, including the details of the hybrid model and the model optimization algorithm. Section 4 presents the construction method of the multi-person fall detection test dataset and gives the experimental protocol and results to verify the effectiveness and feasibility of the proposed scheme. Finally, the paper discusses the conclusions and directions for future work.

II. Related Work

Human fall detection is an independent human action recognition research direction, and researchers have proposed various methods for different technical characteristics. We focus on two issues: the design of a human fall detection model for multi-person and the optimization scheme of the model.

A. Human Fall Detection

In Fig.2, multi-person fall detection can be seen as an extension of single-person fall detection technology, and fall detection belongs to the human action recognition research field. Currently, three main fall detection methods exist 1) Environmental-device fall detection approaches. Detection is based on the environmental noise formed when the human body falls, such as sensing changes in object pressure and sound to detect falls [8]. This method has a high false alarm rate and cost, which is rarely used. 2) Wearable-sensor fall detection approaches. Falls are detected using accelerometers and gyro-scopes [9]. This method requires a long time to wear sensors, which not only affects the comfort of human life but also increases the burden on the body of the elderly. The false alarm rate is high in complex environments. 3) Computer-vision fall detection approaches [10]. It can be divided into two categories: the traditional machine vision method extracts fall features, has low hardware requirements but is susceptible to environmental factors such as background and light changes, and has poor robustness. The other category is artificial intelligence methods, which use camera image data to train and infer convolutional neural networks. This type of solution has the features of high recognition accuracy, no perception, and low cost.

In vision-based human fall detection schemes, human information from multiple modalities can be used as features of the model, such as appearance, depth, optical flow, and human skeleton [11]. Among them, the human skeleton node usually complements other class of modal features, which can convey important information and performs better in model accuracy and robustness [12]. Human skeletal node data mainly contains two dimensions of information, the temporal dimension and the spatial dimension [13]. In this case, the temporal dimension is information about the nature of the action being performed and how it is being performed. We review approaches to such modeling, most of which rely on RNN (Recurrent Netural Network) or convolutional neural network (CNN). WRNNs are neural

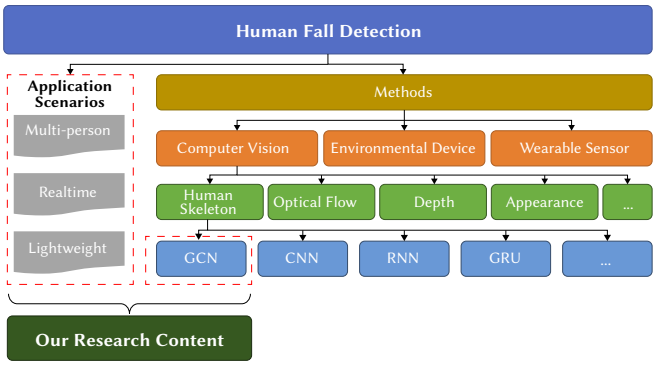


Fig. 2. Human fall detection method.

networks designed to process each time step of a time series one after the other, thus allowing the processing of variable length sequences. They maintain an internal state that captures the temporal context of the signal. RNNs are mostly based on long short term memory (LSTM) or gated recurrent units (GRU). Since the memory unit remembers values at arbitrary time intervals, it enables LSTMs to efficiently capture short-term and long-term temporal dependencies. In fact, LSTM have been widely used in human action-related problems, such as in action recognition [14], [15]. GRU, on the other hand, uses fewer parameters, and therefore less memory and lower computational cost [16], and therefore trains faster than LSTM. However, as shown by Weiss et al [17], LSTM outperforms GRU because it can easily perform unbounded counting, while GRU cannot. Thus, LSTM seems to be more accurate than GRU on longer sequences. In conclusion, the choice between LSTM and GRU depends on the data being processed and the application being considered.

On the other hand, spatial dimensionality, which is used to learn spatial correlation information in skeletal data [13], is modeled by three main categories: spatially- structured architecture (SSA), CNN, and graph convolutional network (GCN).

Among them, SSA relies on network architectures built around the human skeleton to help the model learn spatial correlations and allow the network to compute functions that essentially encode human skeletal features. These methods segment the skeleton into body parts and process the corresponding data in parallel network branches or hierarchical structures. In parallel approaches, the main distinction is usually related to the goal task, which determines the architecture of each branch [18]. Hierarchical approaches, on the other hand, model the human skeleton in layers, which can be either top-down or bottom-up [19], [20]. CNN is another type of architecture that relies on 2D convolution, i.e., in both the spatial and temporal domains. For this reason, the graph structure of the human skeleton is spread along the spatial dimension. CNN is particularly effective in learning spatial correlations in structurally regular data such as images. However, learning the spatio-temporal dynamics of human joints remains a challenge for CNNs because the graph structure of the human skeleton cannot be meaningfully flattened along a single dimension. The researchers have made some optimizations for this as well [21], [22].

Finally, as an extension of CNN, GCN have shown substantial performance advantages [23]. The GCN-based action recognition algorithm models human skeletal nodes as spatial-temporal relationships. It uses graph coarsening and partition design to enable GCN to process non-Euclidean data as efficiently as the human skeleton nodes and can achieve significant performance [24]. Generally, GCNs follow two major branches: Spectral GCN and Spatial GCN [25]. Spectral GCN implements graph con-volution for human skeleton-based action recognition by converting the graph from the

time domain to the frequency domain using the eigenvalues and eigenvectors of the graph Laplacian matrix [26] at the cost of extensive computation. Therefore, several measures are needed to reduce the computational cost of feature decomposition. In contrast, Spatial GCN has a lower computational cost and better performance, which has led to their more comprehensive application [27]. Therefore, most GCN-based approaches in human action recognition have focused on Spatial GCN. Human action is a continuous process, so time is crucial for representing human actions. Since Yan et al. proposed a spatial-temporal graph convolutional network (ST-GCN) in 2018, it has become a research hotspot [28]. Researchers have also proposed various improved versions of ST-GCN schemes. Peng [29] constructed a graph-based search space to explore the spatial-temporal connectivity relationships between nodes for action recognition. Shi [30] proposed that the adaptive learning graph structure is trained and updated along with the model parameters, which are better adapted to the action recognition task. Zhang [31] made the action recognition network more robust by introducing an attention mechanism. Cai [32] constructed a dual-stream model that combines the human pose skeleton and joint-centered lightweight information to capture the local delicate motions around each joint to improve the accuracy of action recognition.

In the multi-person fall detection scheme, [33], [34] used Long Short Term Memory (LSTM) for real-time multi-person fall detection and solved problems such as multi-person occlusion by using multiple cameras. Xu [35] improves the recognition accuracy for multiple users by using multiple trackers. Saturnino [36] proposes a hybrid fall detection model based on YOLO and SVM to improve the model's performance for multiple human targets detection.

Overall, the GCN-based schemes have higher inference accuracy in continuous actions sequences due to the inclusion of spatial and temporal feature information. However, compared to other schemes, GCN-based schemes generally have larger model sizes and do not have special treatment for multi-person scenarios, which is the issue we focus on.

B. Model Optimization

With the rise of Edge AI, more and more intelligent application scenarios occur at the edge end. In particular, some applications with high real-time requirements require the system to respond promptly in the production environment of the data [37]. How-ever, there is a significant contradiction between the vast scale of AI models and the constrained resources of edge devices [38]. While continuously improving the performance of Edge AI devices, accelerating the model inference rate through model optimization techniques is also the key to solving this problem [39], [40]. Among them, efficient network architecture design and model compression are typical approaches [41].

The modules are connected in the efficient network architecture design approach by creating a compact neural network structure and carefully designing the topology [42]. The goal is to achieve efficient deep learning models with acceptable accuracy while ensuring small model structure (low memory) and low computational complexity (high speed). For example, MobileNet and MobileNetV2 use deeply separable convolutional modules. In contrast, separable convolution with residuals/reverse residuals is its basic building block, significantly reducing the parameter size and achieving higher accuracy [43]. Similar examples of building blocks of efficient neural networks to reduce parameters and improve efficiency include ShuffleNetV1 and ShuffleNetV2 [44].

Unlike the efficient network architecture design approach, the model compression method aims to modify a given neural model to reduce its storage and computational costs [45]. In Fig. 3, the neural network-based model compression methods include pruning, quantization, low-rank factorization, and knowledge distillation. Among them, pruning is one of the powerful techniques to remove unimportant components from the model [46]. Pruning is flexible and efficient in removing layers, neurons, connections, or channels, reducing the model by removing redundant parts. The purpose of quantization is to reduce the number of bits required for the model parameters to reduce the cost of storage and computation of the model parameters. Most processors use 32 or more bits to store the parameters of a deep model, which are stored in 16 or 8 bits after processing by quantization [47]. Knowledge distillation and low-rank factorization are also famous and influential in compressing depth models [48], [49].

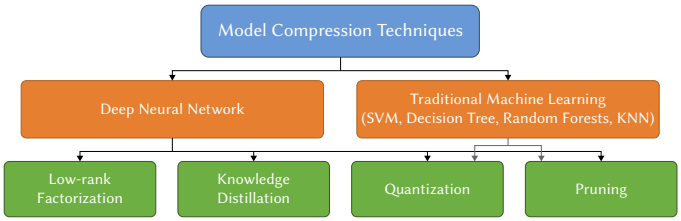


Fig. 3. Model compression method.

III. The Proposed Scheme

In Fig. 4, the traditional human action recognition scheme based on human skeletal nodes includes human posture estimation and human motion recognition. In the human pose estimation phase, the human skeletal node is extracted by human pose detection systems such as Microsoft Kinect and OpenPose and then handed over to the human action recognition model for processing. As mentioned earlier, ST-GCN is based on spatial-temporal information of human skeleton nodes and performs well in long action recognition, so our paper will use ST-GCN in the action recognition phase. ST-GCN generally uses OpenPose to extract skeleton nodes, but OpenPose usually extracts skeleton node information of multiple people from the global level at one time. This method has problems such as poor target recognition accuracy and poor interference resistance between skeleton nodes. In response, this paper optimizes the multiple human target detection process by adding the object detection model YOLO before the human pose estimation phase. The scheme starts with YOLO processing the original real-time video to generate multiple independent human detection boxes. Then each box is handed over to the human pose estimation model. Finally, the extracted human bones are processed by ST-GCN separately. The system's detection accuracy for multiple human targets can be improved by incorporating YOLO. In addition, by adopting a detection scheme based on small image blocks, the system dramatically reduces the computational load of the pose estimation model. Finally, this paper will accelerate the model inference rate through model optimization techniques.

A. Multi-Person Detection With YOLO

YOLO is an object detection model that uses a prediction method based on the whole frame [50]. After scanning an image only once, YOLO can detect all target information, including category and location, and performs well in object detection tasks. The latest version of the YOLO series is YOLOv7, which is currently the fastest object detection model [51]. Previous versions of YOLO have also differed significantly regarding network structure, parameter size, and model performance. Among them, the YOLOv5 version has a more balanced performance in all aspects and is widely used in various application scenarios. Unlike previous versions, YOLOv5 implements a series of network architectures, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These five models are similar in structure.

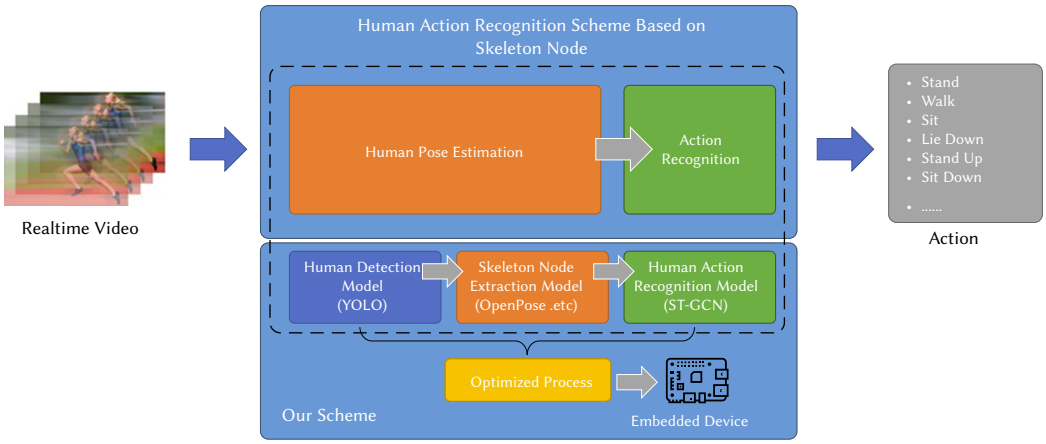


Fig. 4. The main framework of multi-person fall detection scheme.

TABLE I. Comparison of YOLOv5 Series Model

Methods	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
Depth multiple	0.33	0.33	0.67	1.00	1.33
Width multiple	0.25	0.50	0.75	1.00	1.25
C3-n (True)	1,2,3,1	1,2,3,1	2,4,6,2	3,6,9,3	4,8,12,4
C3-n (False)	1	1	2	3	4
Convolution kernels	16,32,64,128,256	32,64,128,256,512	48,96,192,384,768	64,128,256,512,1024	80,160,320,640,1280
Params (MB)	3.90	14.10	40.80	89.30	166.00
Speed (ms)	6	7	14	25	47

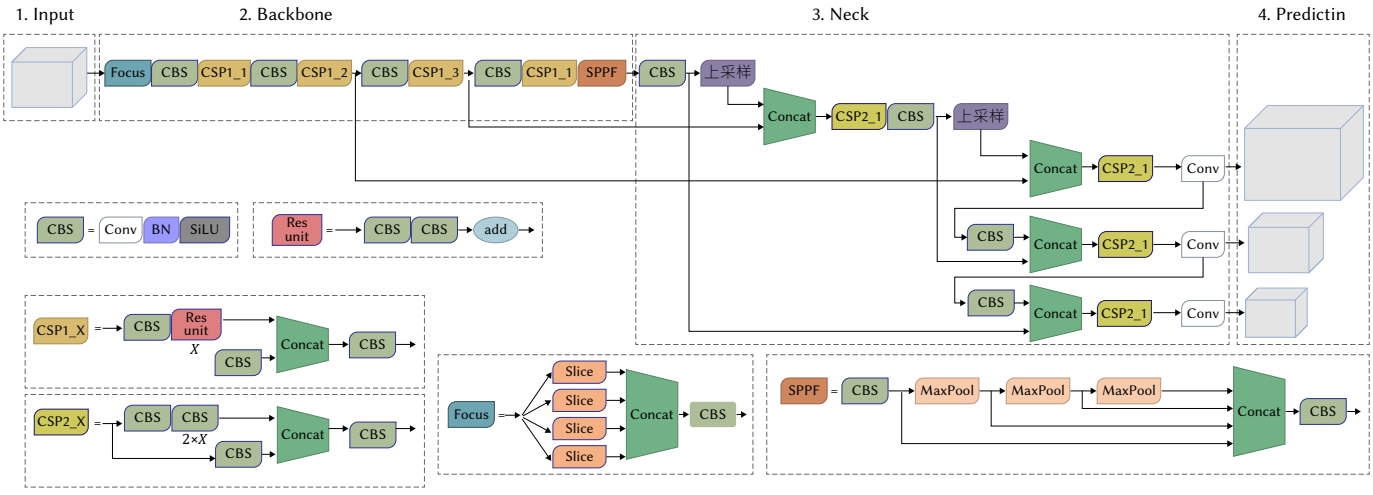


Fig. 5. Illustration of YOLOv5s's network.

The number of convolution kernels in the convolution process can be varied by changing the depth multiplier and the number of C3S in BottleneckC3 (Bottleneck CSP structure with 3 CBS modules) and the width multiplier. It allows for combinations between different network depths and widths to achieve a balance between accuracy and efficiency, as shown in Table I.

In the latest YOLOv5 series, the model size is 3.87MB for YOLOv5n and 14.1MB for YOLOv5s. They are low-cost target detection models suitable for deployment on mainstream mobile or edge devices. In the COCO data set, YOLOv5n was used for object detection, 26.4% of the images had missing person detection, and the time of each image was 6ms on average. YOLOv5s was used for target detection, and 13.8% of the images were missing people detection, and the average time of each image was 7ms. To ensure fall detection accuracy, YOLOv5s is chosen as the base model in this paper for optimization and improvement.

Fig. 5 shows the network framework of YOLOv5s-6.0, which consists of four parts: 1) Convolutional network-based Backbone network, which mainly extracts image feature information. 2) Head detection head, main prediction object box, and prediction object category. 3) The Neck layer between the trunk network and the detection head. 4) The prediction layer outputs the detection results and predicts the object detection frame and label category.

In Fig. 6, this paper obtains video data from the camera and processes each image frame. After YOLO processing, the system will detect multiple human targets. These objects will be given to the human pose estimation model as independent image blocks to extract human skeleton nodes. Since the entire image does not need to be searched during the skeleton node extraction phase, it can be directly based on the independent image block containing the human object, which will significantly improve the inference speed and accuracy of the model.

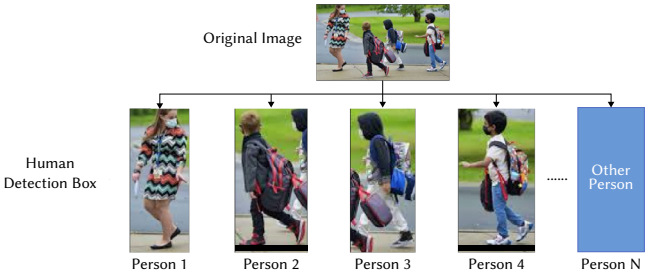


Fig. 6. Human detection box in YOLO.

B. Multi-Person Fall Detection With ST-GCN

ST-GCN pioneered the application of GCN in human action recognition based on skeletal nodes. In Fig. 7, the input to ST-GCN is a joint coordinate vector of a graph node that is obtained based on one graph node and two graph edges. The graph nodes are the skeletal nodes, and the graph edges are the spatial and temporal edges of the skeletal nodes. One of these is the spatial edge between the different skeletal nodes, representing the skeletal constraint information of the human action. The other category is the temporal edges, which are connected between the same skeletal nodes at different moments and represent the temporal constraint information of the human action. These data extract high-level features through the spatial-temporal graph convolution operation. Then the corresponding action classification is obtained as the output using the SoftMax classifier. ST-GCN integrates the temporal and spatial information of skeletal nodes in human actions and performs well in long-action recognition. In this paper, we use ST-GCN for human action recognition.

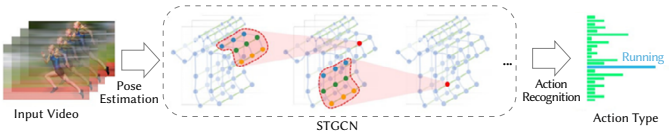


Fig. 7. Illustration of ST-GCN.

The principle and processing flow of the multi-person fall detection method designed in this paper is similar to that of the single-person scheme. In Fig. 8, YOLO identifies multiple human targets from video images and sends them to the human pose estimation model for processing in the form of a human image detection frame. The human posture estimation model extracts bone nodes from each image detection box and further generates a continuous bone image sequence, and ST-GCN will use this for action recognition. Such a scheme that separates the object detection part from the action recognition makes our scheme able to improve the recognition accuracy and speed of the whole system for the actions of multiple people just by adopting the object detection model similar to YOLO.

ST-GCN is a human action recognition model based on skeleton nodes, which can be extracted using OpenPose. OpenPose is a convolutional neural network based on Caffe, a real-time 2D multi-person pose estimation model developed by Carnegie Mellon University (CMU) [52]. It is a bottom-up human pose estimation model which can realize the pose estimation of human movement, facial expression, and

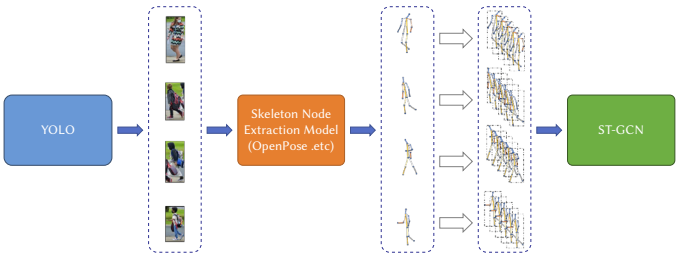


Fig. 8. The main framework of our scheme.

finger movement. It is suitable for single and multi-person scenarios, with excellent recognition effects and fast recognition speed. Fig. 9(a) shows that OpenPose commonly uses the skeletal model containing 25 nodes. Dot 0 represents the nose; Dot 15 to 18 represents the left and right eyes and ears; Dot 1 represents the neck; Dot 2 to 7 represents the left and right shoulders, elbows, and wrists; Dot 8 represents the center of the buttocks, dot 9 to 14 represents the left and right hips, knees and ankles; Dot 19 to 24 represents the left and right nodes of the feet, toes, and heels. To improve the processing efficiency of the model, we simplified the skeletal node structure, deleting one hip center node, 4 facial nodes, and 6 feet nodes. Finally, our skeletal model contains only 14 nodes, as shown in Fig. 9(b).

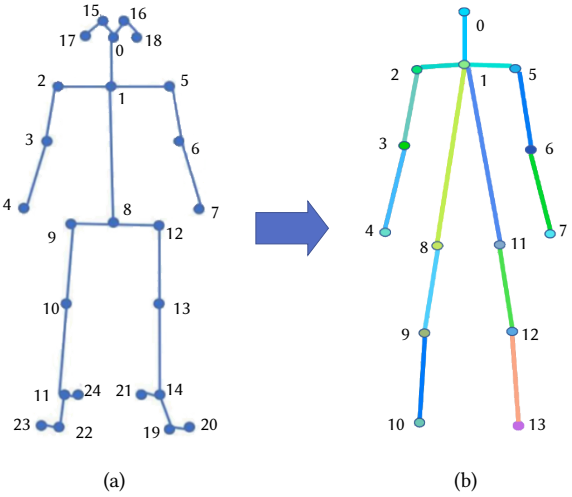


Fig. 9. Illustration of skeleton node structure.

Fig. 10 shows the simplified human skeleton nodes extracted by OpenPose. It can be seen that this method can accurately identify bone nodes in different scenarios. Both (a) and (b) are RGB images, and (c) is the depth image. Among them, the illumination condition of (a) is poor, and the illumination condition of (b) is better. After each frame, OpenPose will generate a 3*14 skeleton node matrix M. The M3*14 stores 14 skeleton nodes, and each node data contains a set of (X, Y) image coordinate information and the confidence Score of the node. The higher the value of the Score, the higher the accuracy of the predicted skeleton node.

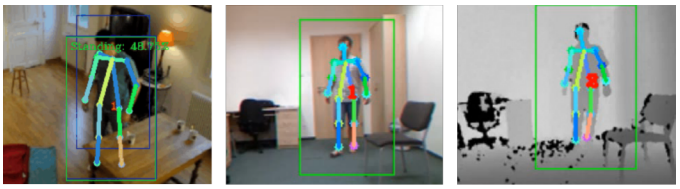


Fig. 10. Skeleton extraction in different scenes.

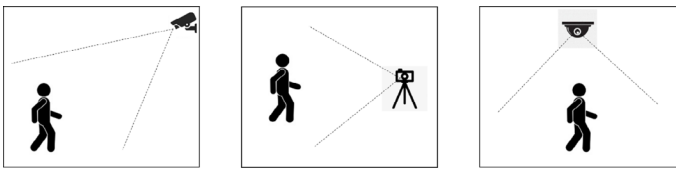
TABLE II. COMPARISON OF FALL DETECTION DATASETS

Dataset	Type	Number of samples	Remark
MCFD	Video	Total 24 videos 1) 14 fall videos 2) 10 non-fall videos	1) Each video contains eight videos from different perspectives
Le2i FDD	Video	Total 191 videos, totaling 75911 frames 1) 132 fall videos, totaling 43745 frames 2) 59 non-fall videos, totaling 32166 frames	1) 1 overhead camera angle 2) Coffee-Room and Home scenes are labeled 3) Lecture-Room and Office scenes are unlabeled
FDD	RGB Image Depth Image	Total 22636 images 1) 4212 fall images 2) 18424 non-fall images	1) 8 overhead camera angles 2) 5 different rooms
URFD	RGB Image Depth Image Video	Total 70 videos, totaling 14539 frames 1) 30 fall data sets, totaling5990 frames 2) 40 daily activity data sets, totaling 8549 frames	1) 2 camera angles are parallel to the ground and directly above the ceiling 2) 1 camera angle records average activity data, and the position is parallel to the floor

There are many kinds of fall detection datasets based on RGB images [53]. Standard lightweight datasets include the Multiple Cameras Fall Dataset (MCFD), Le2i Fall Detection Dataset (Le2i FDD), and the University of Rzeszow Fall Detection (URFD) Dataset. The details are shown in Table II. MCFD [54] contains 24 action sequences recorded by 8 cameras from different angles. The same subject performed the fall action and Activities of Daily Living (ADL), recording ten action types. Le2i FDD [55] used a single RGB camera, and 9 subjects performed 3 types of fall actions and six different ADLs. The videos were captured in 4 different environments (home, coffee room, office, and lecture hall). Actions are carried out in various factors such as light, clothing, the color of the dress, texture of clothing, shadows, reflections, camera view, etc. FDD [56] was recorded from 8 different viewpoints in 5 rooms. The study had five participants, including 2 males and 3 females. The actions performed by the participants included standing, sitting, lying down, bending over, and crawling, which were recorded at a rate of 30 images per second. URFD [57] was produced by the Interdisciplinary Center for Computational Modeling at Rzeszow University. The video sample contains 70 action sequences recorded at 30 frames per second. The dataset recorded falls and ADL, such as standing, bending, and lying down. The environment has adequate lighting. Although these datasets only contain single-person samples, since the fall detection model for multiple people designed in this paper is based on the single-person action recognition method, it can be used as a training dataset for ST-GCN.

In Fig. 11, the positions of the cameras of the dataset can be classified into three types depending on the application scenario. Among them, the height of the camera position in 10(a) is about 45 degrees of the elevation angle of the human line of sight. The device's height in 10(b) is about the waist of the human body. The equipment of 10(c) is located at the top of the ceiling. Generally, scheme (a) is more commonly used [58], [59]. Still, to avoid the influence of the impression model on action recognition due to the camera position factor, the samples of the training data set in this paper will be obtained from FDD and URFD [60]. We augment the original dataset with examples from these two datasets using image data augmentation methods such as symmetry inversion, motion blur, brightness change, and image rotation. After expansion, FDD contains 1084 groups, URFD has 847 groups, the sample resolution is 640*480.

We mix the two augmented datasets, on the one hand, to ensure a sufficient number of samples and, on the other hand, to increase sample diversity and avoid overfitting. In Table III, we extract a certain amount of fall action and non-fall action training samples from each dataset to form the Mix-Dataset of this paper for model training.



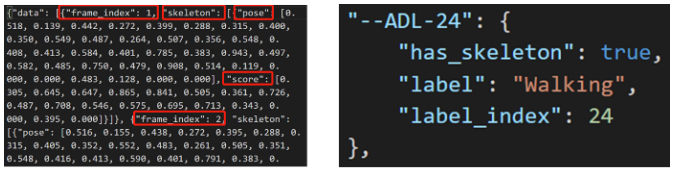
(a) (b) (c)

Fig. 11. Camera location.

TABLE III. THE DATASET COMPOSITION

Dataset	Fall Action (Groups)	Non-Fall Action (Groups)
FDD	550	300
URFD	380	270
Mix Dataset	930	570

The Mix Dataset samples are all images that can be directly used to train CNN-based and RNN-based fall detection models. However, the GCN-based scheme processes the action sequence data based on time series, and the images need to be further processed into the kinetics-skeleton format required by ST-GCN. The kinetics-skeleton converts a sequence of skeletal actions into a list $V_{List} = \{V_1, V_2, \ldots\ldots V_n\}$. Each V_k represents one image, which consists of three parts. $V_k = \{frame\_index{=}k,\ skeleton\ \{pose[p_1, p_2, \ldots\ldots p_m],\ score[s_1, s_2, \ldots\ldots s_p]\}\}$. Fig. 12(a) shows that the *frame_index* represents the frame number in the action sequence. The *pose* represents the point information in the transformed graph vector of the current skeleton node, which indicates the state of the skeletal node. The *score* represents the edge information in the transformed graph vector of the current skeleton node and indicates the state between skeleton nodes. Fig. 12(b) is the label information of this skeleton sample. In our paper, we set $n = 150$, $m = 35$, and $n = 17$. It is expressed as 300 frames per training sample, about 10 seconds per video calculated with 30 frame/s.



(a) (b)

Fig. 12. Kinetics-skeleton node format.

C. Model Optimization

To further improve the real-time performance of the proposed scheme, we use two methods to optimize the model. In particular, we optimize the YOLO by enhancing the model network structure and use the AI acceleration framework-TensorRT to reoptimize the YOLO and ST-GCN.

1. Optimized Design of YOLO

The CSPDarkNet53 backbone network used in YOLOv5s is a Cross Stage Partial Network (CSPNet) introduced in Darknet53 [23] to extract sufficient depth feature information. To enable the YOLO with a more robust feature extraction capability, we used MobileNetV3 to attempt to achieve a coordinated balance of lightweight, accuracy, and efficiency. MobileNetV3 is a light backbone network that performs better on the edge and mobile side [61].

MobileNet (i.e., MobileNetV1) is a lightweight CNN, which is more suitable for deployment on the edge devices. It can use Depthwise Separable Convolution (DSC) to vary the computation of convolution to reduce the number of network parameters to balance accuracy and efficiency. MobileNetV2 adds two new modules: Reverse Residuals (IR) and Linear Bottlenecks (LB). The IR module can make the model have better feature transmission capability and deeper network layer. Meanwhile, MobileNetV2 uses LB module instead of the non-linear module, thus reducing the loss of the model on low-level features. MobileNetV3, released in 2019, combines some of the structures from V1 and V2 and removes the more computationally expensive network layer from the V2 architecture. It achieves low resource consumption while guaranteeing almost no loss of accuracy by introducing the lightweight attention structure of Squeeze and Excitation Networks (SE-Net) [62].

In Fig. 13, we replace the backbone of YOLOv5s with the feature extraction network of MobileNetV3. In the YOLOv5s, three different sizes of feature maps can be obtained after three down-sampling, and then feature fusion is performed. The locations of the three down-samplings are identified with a,b,c. We choose the output of the last three down-samplings of the MobileNetV3 feature extraction network as an alternative. Specifically, the feature extraction network of MobileNet3 contains 13 sampling modules Module [0-12]. Among them, Module[4] is the penultimate third down-sampling, so Module[0-3] is used as MobileNet1, whose down-sampling position is identified with 1; Module[9] is the penultimate third down-sampling, so Module[4-8] is used as MobileNet2, whose down-sampling position is identified with 2; and, finally, Module[9-12] as MobileNet3, whose down-sampling position is identified with 3. The results of these three down-samplings will be used for subsequent processing in YOLOv5s.

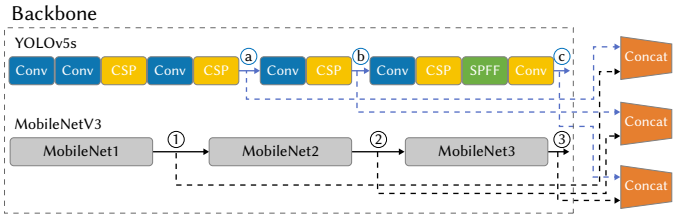


Fig. 13. Depthwise separable convolution framework.

DSC consists of Depthwise Convolution (DW) and Pointwise Convolution (PW) [63], as shown in Fig. 14, and the parameters and computational effort of DSC are significantly reduced compared to traditional convolution. A comparison of the computational effort between the two is shown in (1). W1 and W2 are the computational costs of DSC and the standard conventional convolution, respectively. The size of the convolution kernel for MobileNetV3 feature extraction is mainly 5×5. Therefore, the computational cost of DSC is about 1/25 of the traditional conventional convolution.

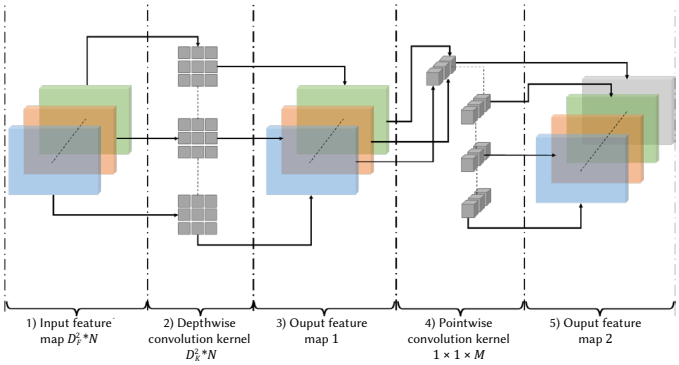


Fig. 14. Depthwise separable convolution framework.

$$\frac{W_1}{W_2}=\frac{D_K^2\times M\times D_F^2+M\times N\times D_F^2}{D_K^2\times M\times N\times D_F^2}=\frac{1}{N}+\frac{1}{D_K^2} \tag{1}$$

The Fig. 15(a) is the residual network structure, and Fig. 15 (b) is the reverse residual network structure. Reverse residual networks use point convolution to expand the number of channels, then deep convolution in higher layers, and finally, use point convolution to shrink the channels. Reverse residual networks improve the gradient propagation of features with the help of residual connections, making the network layer deeper. The network uses minor input and output dimensions, significantly reducing the network's computational consumption and parameter size. In addition, the reverse residual network is CPU and memory efficient for inference and enables the construction of flexible mobile-side models, making it suitable for edge side applications.

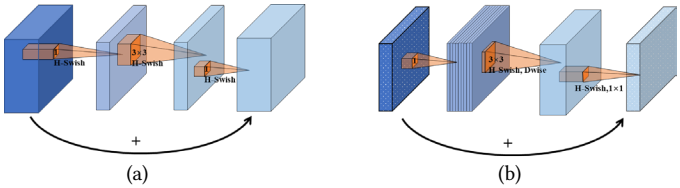


Fig. 15. Residual network and reverse residual network.

MobileNet includes two hyperparameters, α, and β. Where α represents the width factor, which can be adjusted to α times the original convolution kernel by adjusting the number of convolution kernels, and β represents the control input image size. In this paper, the amount of computation after adjusting α using DSC in (2).

$$W=D_K^2\times\alpha M\times D_F^2+\alpha M\times\alpha N\times D_F^2 \tag{2}$$

By adjusting the α, the calculation effort and model volume can be directly reduced to $1/\alpha^2$, which significantly reduces the model's number of parameters and computational effort with minimal loss of accuracy. We set α=0.5, and the optimized model is YOLOv5s-opt. In this paper, SSD, Faster-RCNN, YOLOv4, YOLOv5s, and Yolov5s-opt are tested on the COCO dataset, and the results are shown in Table IV. It can be seen that the optimized YOLOv5s-opt reduces the parameter size by 50% and improves the frame rate by 15% compared with YOLOv5s.

2. Optimized Design of ST-GCN

In this paper, TensorRT will be used to optimize ST-GCN. It is an AI optimization and deployment framework designed by NVIDIA for GPU [64]. In Fig. 16, it is both an inference optimization engine and a runtime execution engine. It provides optimal support for the model's inference at the graphics optimization, operator optimization, memory optimization, and Int8 calibration levels. Specifically, it benefits from the fact that after training the neural network, TensorRT can compress,

TABLE IV. Performance Comparison of Object Detection Models

Model	Params (MB)	FPS (frame/s)
SSD	100	67
Faster-RCNN (ResNet50)	109	42.9
YOLOv4 (CSPDarkNet53)	24.5	31.5
YOLOv5s (CSPDarkNet53)	14.1	61.5
YOLOv5s-opt (ours)	7.2	70.2

optimize, and deploy the network at runtime without the overhead of a framework. It can also improve the latency, throughput, and throughput of the network through combining layers, kernel optimization selection, as well as performing optimization and conversion to optimal matrix math methods based on a specified precision.

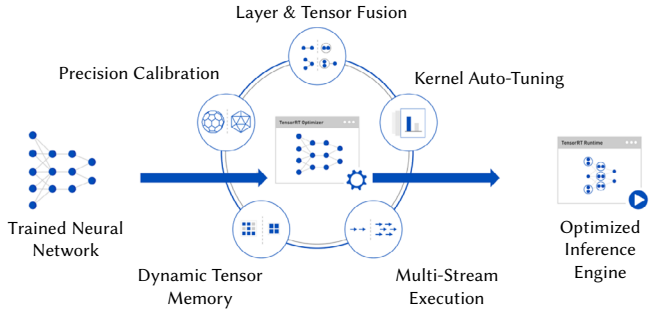


Fig. 16. Model optimization processing flow of TensorRT.

The operation of TensorRT consists of two main phases, Build, and Deployment. The Build phase involves the conversion of the model from another model form to a TRT form. During the model conversion, the inter-layer fusion and accuracy calibration of the optimization mentioned above is completed. The output of this step is an optimized TRT model for the specific GPU platform and network model, serialized to disk or memory in the form of a plan file. The plan file from the previous step is first deserialized, a Runtime Engine is created, and the inference task can then be executed. The YOLO and ST-GCN are built in the PyTorch framework and converted into TRT models using ONNX intermediate conversions [65], as shown in Fig. 17. After optimization, the model can reduce the model's parameters by 16%.

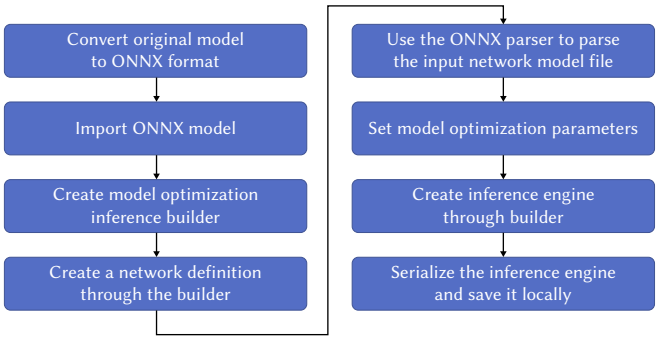


Fig. 17. The optimization process of YOLO and ST-GCN.

IV. Experiments and Discussions

Experimental hardware: CPU: 11th Gen Intel Core (TM) i7-11700 @2.50 GHz. Memory: 16GB; GPU: NVIDIA GeForce GTX 1080 Ti. GPU Acceleration Library: CUDA 11.0.3, CUDNN 8.2.1. OS: Windows 10 (64-bit). Software tools: OpenCV 4.1.1, Pytorch 1.7.0, TensorRT 7.1.3.

A. Dataset

The fall test datasets in this paper are divided into two types: single-person and multi-person. Among them, the original single-person dataset was selected from the publicly available dataset Le2i FDD with 155 human fall videos, which include 95 ADL videos and 60 fall videos. Each fall action had a complete fall process containing the other action to fall.

In Fig. 18, the scenes include 3 types: home, office, and pantry. The home scene (a) is a living room scene, including sofas, dining tables, stairs, chairs, table lamps, and other accessories, and contains a variety of lighting conditions. The office scene (b) includes tables and chairs with a more regular and balanced light distribution. The pantry scene (c) includes sofas, tables, tea sets, etc. The light is more frequent and evenly distributed. The video resolution is 320×240.

Fig. 18. Original single-person fall detection test dataset.

To experiment more effectively, we created our multi-person fall dataset (MPFDD), which has two scenarios for 2 to 5 persons, respectively. The original dataset consisted of 220 videos divided into indoor and outdoor scenes. It consists of 80 ADL videos and 140 fall videos. The indoor scene is the action room scene, which includes chairs, tables, computers, and other accessories. The outdoor scene is open, with tables and chairs as the main accessories. Both scenes of 2-person and 3-person have good lighting conditions, but it is no need in 4-person and 5-person. In addition, the 2-person scene includes 20 ADL videos, 10 fall videos with 1-person, and 10 fall videos with 2-person. The 3-person scene have 20 ADL videos, 10 fall videos with 1-person, 10 fall videos with 2-person, and 10 fall videos with 3-person. The 4-person scene have 20 ADL videos, 10 fall videos with 1-person, 10 fall videos with 2-person, 10 fall videos with 3-person, and 10 fall videos with 4-person. The 5-person scenes have 20 ADL videos, 10 fall videos with 1-person, 10 fall videos with 2-person, and 10 fall videos with 3-person, 10 fall videos with 4-person, 10 fall videos with 5-person. The video resolution is 1060×510. In Fig. 19, we show some of the videos in the original MPFDD.

Similar to the train dataset, the original single-person dataset and multi-person dataset were expanded using image data augmentation techniques. The main data augmentation algorithms we use include symmetric flipping, adding Gaussian noise, motion blur and brightness contrast transformation.

In Fig. 20, (a) represents the original video, and (b)-(e) represent the effects after being processed by the above four data augmentation algorithms, respectively. We do not process every original video with all four image data augmentation algorithms. In particular, we do not use the brightness contrast transform algorithm for videos that are not very bright. In the end, we get the expanded dataset, which consisted of a total of 1044 videos, which included 413 ADL videos and 631 fall videos.

B. Evaluation Metric

In classification tasks, the confusion matrix is often used to show the results predicted by the model. In this case, the actual classes are represented by the columns of the matrix, while the rows indicate the predicted classes [66]. For each class, the matrix displays true positive

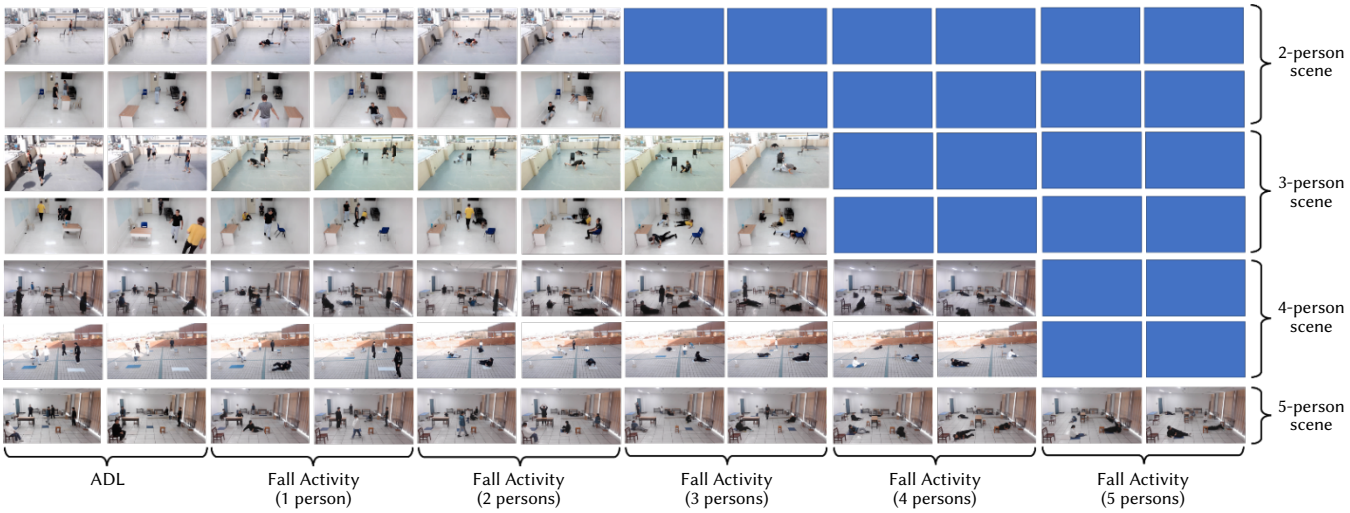


Fig. 19. Original Multi-person fall detection test dataset (MPFDD).

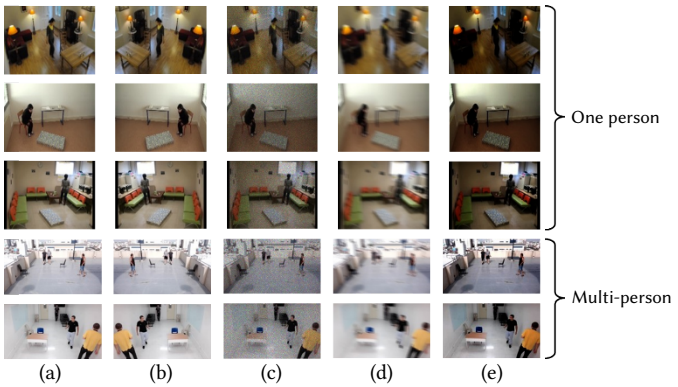


Fig. 20. The dataset expansion operation.

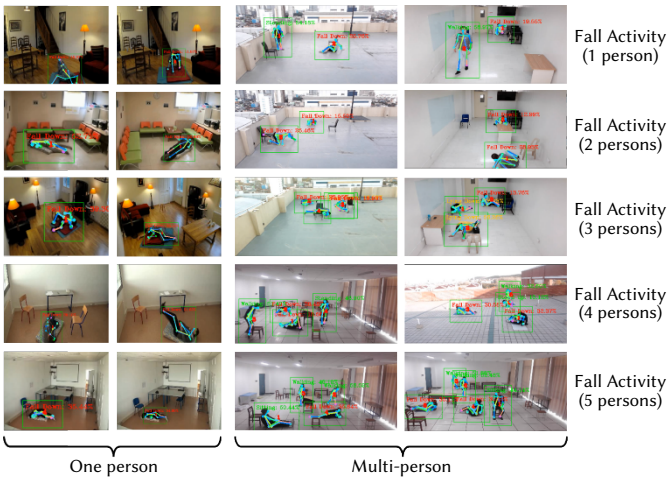


Fig. 21. Multi-person fall detection.

(TP), false positive (FP), false negative (FN), and true negative (TN) values. Where TP is the number of fall targets correctly detected. FP is the number of targets that were falsely detected by falls. FN is the number of samples where falls were not detected. TN is the number where no falls were correctly detected. The confusion matrix can calculate many model evaluation metrics, including precision, recall, accuracy, and F1 score [43], as shown in (3), (4), (5) and (6). In addition, there is the criterion FPS which measures the real-time performance of the model. It indicates the number of images processed per second. In this paper, the models are evaluated according to the above criteria.

$$Precision = \frac{T_P}{T_P+F_P} \times 100\% \tag{3}$$

$$Recall = \frac{T_P}{T_P+F_N} \times 100\% \tag{4}$$

$$Accuracy = \frac{T_P+T_N}{T_P+F_N+T_N+F_P} \times 100\% \tag{5}$$

$$F_1 = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{6}$$

C. Results and Analysis

Fig. 21 shows the results of the proposed scheme for multi-person fall detection. There are various fall detection schemes, mainly including CNN-based method, RNN-based method, GCN-based method, and some other types of methods. We select two typical methods in each type of scheme to compare with our proposed scheme. The comparison results are shown in Table V. Among them, Carlier [67] and Zhang [68] use CNN-based methods. Kareem [69] and Yadav [70] use RNN-based methods. Zheng [71] and Lee [72] use GCN-based methods. The above methods are all estimate human skeleton nodes as features. In addition, Maheswari [73] and Kiran [74] are representatives of two other types of schemes. Maheswari uses the human body as the entire modeling object and the settlement results of HOG and SRMAR as features for detection. Kiran extracts human behavioral features at different stages through a combination of multiple CNNs, and introduces SVM models for inference of the results. We also use an optimization scheme based on ST-GCN, so it is also incorporated. Miss detection represents the number of samples in the fall video that fails to detect fall action, and false detection represents the number of samples in the fall test video that detect non-fall action as a fall action.

In Table V and Fig. 22, we can see that: 1) the CNN-based scheme is low in terms of accuracy, while the RNN- and GCN-based schemes are relatively high, but the former has an advantage in terms of frame rate. This is because the CNN-based scheme uses prediction based on the spatial information of a single skeletal node. In contrast, the latter two are based on processing a sequence of skeletal nodes with additional temporal information. As a result, the former is less computationally intensive and faster, while the latter has higher detection accuracy for predicting long movements. 2) Our scheme performs better in all metrics compared to other schemes. Our scheme has a less obvious advantage in terms of accuracy since various optimized versions of pose estimation models have been proposed in recent years to improve the accuracy of extracting skeletal nodes. The scheme represented by

TABLE V. Performance Comparison of Different Algorithms

Paper	Method	Miss Detection	False Detection	Precision (%)	Recall (%)	Accuracy (%)
[67]	**Optical-Flow based CNN**	64	25	75.9	74.3	74.7
[68]	**VARNN+CNN**	41	29	78.0	77.2	78.0
[69]	**Optimized RNN**	40	19	81.8	80.1	81.4
[70]	**ConvLSTM**	35	14	82.4	81.6	82.2
[71]	**Optimized GCN**	43	20	82.9	80.4	81.1
[72]	**2s-AGCN**	19	14	86.5	85.7	86.2
[73]	**HOG+SRMAR**	44	29	79.5	78.0	79.2
[74]	**Multi-CNN+SVM**	57	45	73.5	71.2	72.0
--	**ST-GCN**	36	31	81.2	79.8	80.8
--	**Ours**	28	17	84.3	82.7	83.8

[72] uses a two-stream algorithm (2s-AGCN) that can model both first-order and second-order information, significantly improving the recognition accuracy. In comparison, the overall testing performance of both [73] and [74] can reach over 70%, and the inference speed of these two methods is also higher, mainly due to the smaller scale of the model. The model structure used in [74] is relatively simple, and the ability of CNN to extract deep-seated human motion features is limited, and it cannot integrate temporal and spatial information of motion, so its inference accuracy is low. In summary, our method can achieve good performance and speed trade off.

The second and third columns of Table V show that all schemes in the experiment have miss detection and false detection. We analyzed the reasons for these situations. In the experiment, we found that the causes of miss detection and false detection in the multi-person scene are similar to those in the single-person scene. We mainly use the video display effect of single-person scenes to simplify the analysis content and process. The comparison model includes [67], [69], [71], ST-GCN, and ours.

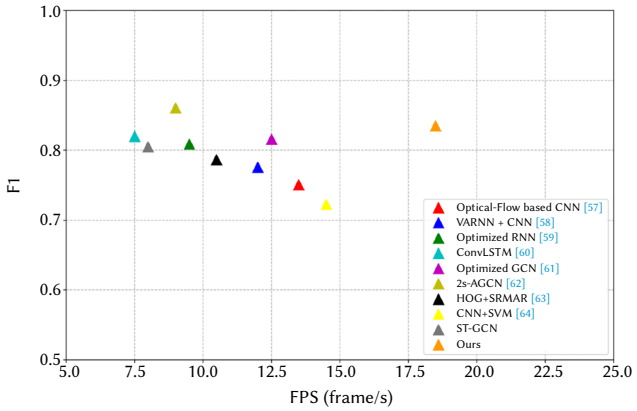


Fig. 22. Multi-person fall detection.

In Fig. 23, the main reasons for miss and false detection can be summarized into four categories, and the red "□" represents the human position in the video. 1) The part of the human body is close to the edge of the image, resulting in the partial loss of the part of the human body image, which is not conducive to the extraction and discrimination of human skeleton features. In (a), most of the head and upper body of the tester are outside the image range, and all test models fail to detect them. 2) Due to the uneven distribution of light in the scene, the gradient features of the human body are not apparent, which affects the extraction of skeletal nodes. In (b), the clothes of the tester and the ambient background were red, and the ambient luminance was insufficient. Although all test models could detect the human target, the extracted skeletal states deviated severely from reality. 3) Skeletal feature extraction failed due to partial or complete occlusion of the human body parts. This is similar to the first case, where the occlusion can be a stationary accessory or a moving person in the scene. In (c), the test person's head and the right half of the body are obscured by the table, and all test models fail to detect. 4) Due to the perspective effect of the camera, when the test tester falls in a direction parallel to the direction of the camera's vision and when the head is further away from the camera than the feet, the tester's skeletal state is projected onto the image in a state of action similar to ADL, which leads to miss detection. In (d), the tester is in a semi-slumped state parallel to the camera's vision direction, with the skeleton showing a state similar to sitting and bending, and all test models fail to detect it.

In the multi-person scene, it is more evidence that some human targets fail to be detected due to mutual occlusion between human bodies. As shown in Fig. 24, the red "□" represents undetected human targets. In short, the main reason for the miss and false detection is the inaccurate extraction of human skeleton node features caused by various environmental factors, which will provide a clear research direction for our further work.

V. Conclusions and Future Scope

To improve the quality of life of the elderly, we propose a fall detection scheme based on human skeleton nodes. This scheme is a hybrid model based on YOLO and ST-GCN, which can support multiple fall detection. We also use model optimization technology to accelerate the proposed model, further reduce the scale of the model and improve the inference speed. The experimental results show that in a good testing environment, this scheme has high detection accuracy and obvious real-time advantages. Therefore, as an attempt at a hybrid model for multi person fall detection, this scheme has some reference value for subsequent research. Our scheme also has problems because the detection cannot be recognized due to uneven light distribution, blocked human body parts, and unique fall direction. To improve the detection accuracy of the proposed model, the following aspects will be studied in future work:

- Study of light adaptive compensation algorithms. Incorporate it into a fall detection system to increase the system's resistance to light changes.
- Study of multi-camera detection methods. Try using multiple cameras for fall detection from various angles to solve the occlusion problem.

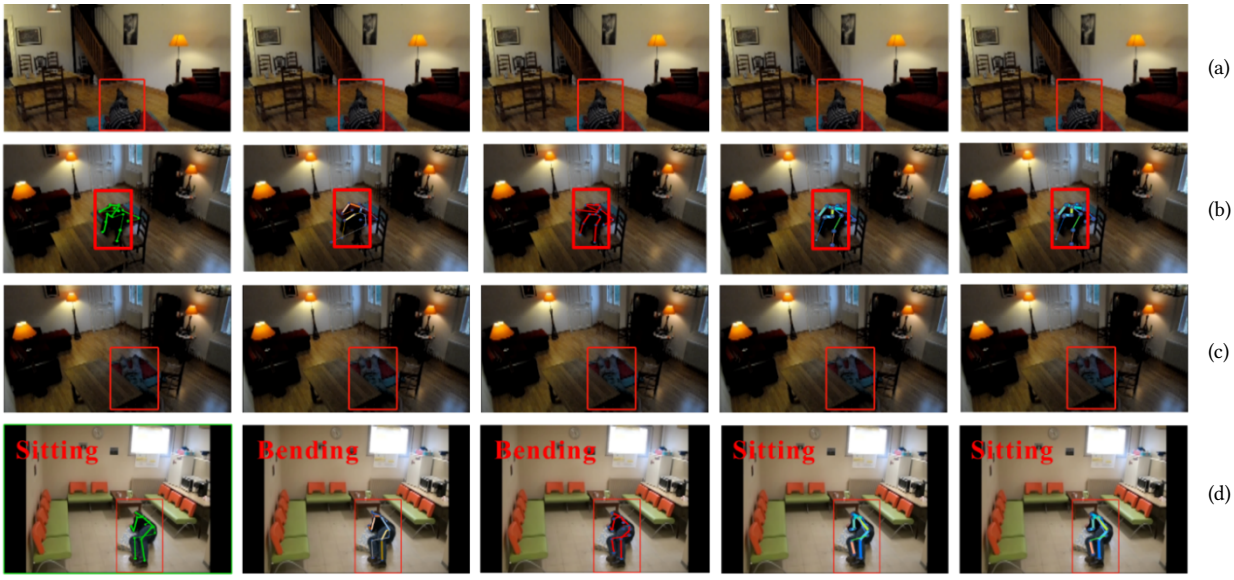


Fig. 23. Miss detection and false detection of different algorithms.

Fig. 24. Miss detection in multi-person scene.

- As there is no publicly available public dataset dedicated to the abnormal action of the elderly, such data is missing from our test dataset MPFDD. In future work, we will gradually collect data on the abnormal action of the elderly through cooperation with relevant medical institutions and elderly care institutions. We will also improve the fall experiment by adding more human actions and test scenarios to optimize the proposed scheme's shortcomings further.

Acknowledgment

This study received support from the following sources: the University Natural Science Foundation of Anhui Province (Grant no. 2023AH051542 and Grant no.2022AH010085).

References

[1] Ageing and health, World Health Organization., 2022. Accessed: June. 8, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health.

[2] World Population Ageing: 1950–2050, Global Action on Aging., NY, USA, 2002. Accessed: June. 8, 2024. [Online]. Available: http://globalag.igc.org/ruralaging/world/ageingo.htm.

[3] S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, "Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review," *Sensors*, vol. 21, no. 15, pp. 5134-5156, 2021, doi: 10.3390/s21155134.

[4] S.-H. Jung, J.-M. Hwang, and C.-H. Kim, "Inversion Table Fall Injury, the Phantom Menace: Three Case Reports on Cervical Spinal Cord Injury," *Healthcare*, vol. 9, no. 5, pp. 492-500, 2021, doi: 10.3390/healthcare9050492.

[5] Falls, World Health Organization., 2021. Accessed: June. 8, 2024. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls.

[6] H. Ramirez, S. A. Velastín, I. Meza, E. Fabregas, D. Makris, and G. Farias, "Fall Detection and Activity Recognition Using Human Skeleton Features," *IEEE Access*, vol. 9, pp. 33532-33542, 2021, doi: 10.1109/ACCESS.2021.3061626.

[7] H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastín, "Video-based Human Action Recognition using Deep Learning: A Review," *ArXiv*, vol. abs/2208.03775, pp. 1-25, 2022, doi: 10.48550/arXiv.2208.03775.

[8] X. Li, J. Li, J. Lai, Z. Zheng, W. Jia, and B. Liu, "A Heterogeneous Ensemble Learning-Based Acoustic Fall Detection Method for Elderly People in Indoor Environment," *Artificial Intelligence in HCI: First International Conference, AI-HCI 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*. Springer International Publishing, 2020, pp. 369-383., doi: 10.1007/978-3-030-50334-5_25.

[9] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "Real-Life/Real-Time Elderly Fall Detection with a Triaxial Accelerometer," *Sensors*, vol. 18, no. 4, pp. 1101-1118, 2018, doi: 10.3390/s18041101.

[10] J. Gutiérrez, V. Rodríguez, and S. Martín, "Comprehensive Review of Vision-Based Fall Detection Systems," *Sensors*, vol. 21, no. 3, pp. 947-996, 2021, doi: 10.3390/s21030947.

[11] R. Josyula and S. Ostadabbas, "A Review on Human Pose Estimation," *ArXiv*, vol. abs/2110.06877, pp. 1-24, 2021, doi: 10.48550/arXiv.2110.06877.

[12] J.-L. Chung, L.-Y. Ong, and M. C. Leow, "Comparative Analysis of Skeleton-Based Human Pose Estimation," *Future Internet*, vol. 14, no. 12, pp. 380-198,.2022, doi: 10.3390/fi14120380.

[13] L. Mourot, L. Hoyet, F. L. Clerc, F. Schnitzler, and P. Hellier, "A Survey on Deep Learning for Skeleton-Based Human Animation," *Computer Graphics Forum*, vol. 41, no. 1, pp. 122-157, 2021, doi: 10.1111/cgf.14426.

[14] W. W. Y. Ng, M. Zhang, and T. Wang, "Multi-Localized Sensitive Autoencoder-Attention-LSTM For Skeleton-based Action Recognition," *IEEE Transactions on Multimedia*, vol. 24, pp. 1678-1690, 2022, doi: 10.1109/TMM.2021.3070127.

[15] S. A. Khowaja and S.-L. Lee, "Skeleton-based human action recognition with sequential convolutional-LSTM networks and fusion strategies," *Journal of Ambient Intelligence Humanized Computing*, vol. 13, no. 8, pp. 3729-3746, 2022, doi: 10.1007/s12652-022-03848-3 .

[16] L. Lu, C. Zhang, K. Cao, T. Deng, and Q. Yang, "A Multichannel CNN-GRU Model for Human Activity Recognition," *IEEE Access*, vol. 10, pp. 66797-66810, 2022, doi: 10.1109/ACCESS.2022.3185112.

[17] G. Weiss, Y. Goldberg, and E. Yahav, "On the Practical Computational Power of Finite Precision RNNs for Language Recognition," *ArXiv*, vol. abs/1805.04908, pp. 1-9, 2018, doi: 10.48550/arXiv.1805.04908.

[18] X. Guo and J. Choi, "Human Motion Prediction via Learning Local Structure Representations and Temporal Dependencies," *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, January 27–February 1, 2019, AAAI Press Publishing, vol. 33, no. 1, pp: 2580-2587, 2019, doi: 10.1609/aaai.v33i01.33012580.

[19] H. Wang, E. S. L. Ho, H. P. H. Shum, and Z. Zhu, "Spatio-Temporal Manifold Learning for Human Motions via Long-Horizon Modeling," *IEEE Transactions on Visualization Computer Graphics*, vol. 27, no. 1, pp. 216-227, 2019, doi: 10.1109/TVCG.2019.2936810.

[20] Y. Li et al., "Efficient convolutional hierarchical autoencoder for human motion prediction," *The Visual Computer*, vol. 35, pp. 1143-1156, 2019, doi: 10.1007/s00371-019-01692-9.

[21] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional Sequence to Sequence Model for Human Dynamics," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, June 18-22, 2018, pp. 5226-5234, 2018, doi: 10.1109/CVPR.2018.00548.

[22] C. Zang, M. Pei, and Y. Kong, "Few-shot Human Motion Prediction via Learning Novel Motion Dynamics," *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, Yokohama, Japan, July 11-17, 2020, pp. 846-852, 2020, doi: 10.24963/ijcai.2020/118.

[23] M. Al-Faris, J. Chiverton, D. L. Ndzi, and A. I. Ahmed, "A Review on Computer Vision-Based Methods for Human Action Recognition," *Journal of Imaging*, vol. 6, no. 6, pp. 46-77, 2020, doi:10.3390/jimaging6060046.

[24] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-Based Action Recognition with Shift Graph Convolutional Network," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 180-189, doi: 10.1109/cvpr42600.2020.00026.

[25] L. Feng, Y. Zhao, W. Zhao, and J. Tang, "A comparative review of graph convolutional networks for human skeleton-based action recognition," *Artificial Intelligence Review*, vol. 55, pp. 4275-4305, 2021, doi: 10.1007/s10462-021-10107-y.

[26] L. Huang, Y. Huang, W. Ouyang, and L. Wang, "Part-Level Graph Convolutional Network for Skeleton-Based Action Recognition," *Proceedings of the AAAI conference on artificial intelligence*, New York, USA, February 7–12, 2020, AAAI Press Publishing, vol. 34, no. 7, pp. 11045-11052, 2020, doi: 10.1609/AAAI.V34I07.6759.

[27] U. Bhattacharya, T. Mittal, R. Chandra, T. Randhavane, A. Bera, and D. Manocha, "STEP: Spatial Temporal Graph Convolutional Networks for Emotion Perception from Gaits," *Proceedings of the AAAI conference on artificial intelligence,* New York, USA, February 7–12, 2020, AAAI Press Publishing, vol. 34, no. 2, pp. 1342-1350, 2020, doi: 10.1609/aaai.v34i02.5490.

[28] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," *Proceedings of the AAAI conference on artificial intelligence,* New Orleans, Lousiana, USA, February 2-7, 2018, AAAI Press Publishing, vol. 32, no. 1, pp. 1-9, 2018, doi: 10.1609/aaai.v32i1.12328.

[29] W. Peng, X. Hong, H. Chen, and G. Zhao, "Learning Graph Convolutional Network for Skeleton-based Human Action Recognition by Neural Searching," *Proceedings of the AAAI conference on artificial intelligence,* New York, USA, February 7–12, 2020, AAAI Press Publishing, vol. 34, no. 3, pp. 2669-2676, 2020, doi: 10.1609/AAAI.V34I03.5652.

[30] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-Based Action Recognition With Multi-Stream Adaptive Graph Convolutional Networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532-9545, 2020, doi: 10.1109/TIP.2020.3028207.

[31] D. Zhang, H. Wang, C. Weng, and X. Shi, "Video Human Action Recognition with Channel Attention on ST-GCN," *Journal of Physics: Conference Series*, IOP Publishing, vol. 2010, no. 1, pp. 012131-012136, 2021, doi: 10.1088/1742-6596/2010/1/012131.

[32] J. Cai, N. Jiang, X. Han, K. Jia, and J. Lu, "JOLO-GCN: Mining Joint-Centered Light-Weight Information for Skeleton-Based Action Recognition," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 2735-2744, doi: 10.1109/WACV48630.2021.00278.

[33] M. Taufeeque, S. Koita, N. Spicher, and T. M. Deserno, "Multi-camera, multi-person, and real-time fall detection using long short term memory," *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications. SPIE,* vol. 11601, pp. 35-42., 2021, doi: 10.1117/12.2580700.

[34] M. Meratwal, N. Spicher, and T. M. Deserno, "Multi-camera and multi-person indoor activity recognition for continuous health monitoring using long short term memory," *Medical imaging 2022: imaging informatics for healthcare, research, and applications. SPIE*, vol. 12307, pp. 64-71, 2022, doi: 10.1117/12.2612642.

[35] T. Xu, J. Chen, Z. Li, and Y. Cai, "Fall Detection Based on Person Detection and Multi-target Tracking," *11th International Conference on Information Technology in Medicine and Education (ITME). IEEE*, pp. 60-65, 2021, doi: 10.1109/ITME53901.2021.00023.

[36] S. Maldonado-Bascón, C. Iglesias-Iglesias, P. Martín-Martín, and S. Lafuente-Arroyo, "Fallen People Detection Capabilities Using Assistive Robot," *Electronics*, vol. 8, no. 9, pp. 915-934, 2019, doi: 10.3390/ELECTRONICS8090915.

[37] Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, and Y. He, "Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application," *Comput. Electron. Agric*, vol. 192, pp. 106586-106604, 2022, doi: 10.1016/j.compag.2021.106586.

[38] A. Singh et al., "Artificial intelligence in edge devices," *Advances in Computers*, vol. 127, pp. 437-484, 2022, doi: 10.1016/bs.adcom.2022.02.013.

[39] R. Poojary and A. Pai, "Comparative Study of Model Optimization Techniques in Fine-Tuned CNN Models," *International Conference on Electrical Computing Technologies Applications*, pp. 1-4, 2019, doi: 10.1109/ICECTA48151.2019.8959681.

[40] R. Mishra, H. P. Gupta, and T. Dutta, "A Survey on Deep Neural Network Compression: Challenges, Overview, and Solutions," *ArXiv*, vol. abs/2010.03954, pp. 1-14, 2020, doi: 10.48550/arXiv.2010.03954.

[41] B. L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485-532, 2020, doi: 10.1109/JPROC.2020.2976475.

[42] T. Choudhary, V. K. Mishra, A. Goswami, and S. Jagannathan, "A comprehensive survey on model compression and acceleration," *Artificial Intelligence Review*, vol. 53, pp. 5113-5155, 2020, doi: 10.1007/s10462-020-09816-7.

[43] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proceedings of the *IEEE/CVF Conference on Computer Vision Pattern Recognition*, Salt Lake City, USA, June 18-22, 2018, pp. 4510-4520, 2018, doi: 10.1109/CVPR.2018.00474.

[44] J. Han and Y. Yang, "L-Net: lightweight and fast object detector-based ShuffleNetV2," *Journal of Real-Time Image Processing*, vol. 18, no. 6, pp. 2527-2538, 2021, doi: 10.1007/s11554-021-01145-4.

[45] J.-H. Kim, S. Chang, and N. Kwak, "PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation," *ArXiv*, vol. abs/2106.14681, pp. 1-5, 2021, doi: 10.48550/arXiv.2106.14681.

[46] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *ArXiv*, vol. abs/1710.01878, pp. 1-11, 2017, doi: 10.48550/arXiv.1710.01878.

[47] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," *ArXiv*, vol. abs/1802.05668, pp. 1-21, 2018,

doi: 10.48550/arXiv.1802.05668.
[48] L. Chen, Y. Chen, J. Xi, and X. Le, "Knowledge from the original network: restore a better pruned network with knowledge distillation," *Complex Intelligent Systems*, vol. 8, pp. 709-718, 2021, doi: 10.1007/s40747-020-00248-y.
[49] Y.-W. Hong, J.-S. Leu, and M. Faisal, "Analysis of Model Compression Using Knowledge Distillation," *IEEE Access*, vol. 10, pp. 85095-85105, 2022, doi: 10.1109/access.2022.3197608.
[50] V. Viswanatha, K. ChandanaR, and C. RamachandraA., "Real Time Object Detection System with YOLO and CNN Models: A Review," *ArXiv*, vol. abs/2208.00773, pp. 1-8, 2022, doi: 10.48550/arXiv.2208.00773.
[51] J. Zheng, H. Wu, H. Zhang, Z. Wang, and W. Xu, "Insulator-Defect Detection Algorithm Based on Improved YOLOv7," *Sensors*, vol. 22, no. 22, pp. 8801-8823, 2022, doi: 10.3390/s22228801.
[52] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 43, pp. 172-186, 2018, doi: 10.1109/TPAMI.2019.2929257.
[53] E. Alam, A. Sufian, P. Dutta, and M. Leo, "Vision-based Human Fall Detection Systems using Deep Learning: A Review," *Computers in biology medicine*, vol. 146, pp. 105626-105664, 2022, doi: 10.1016/j.compbiomed.2022.105626.
[54] E. Auvinet, C. Rougier, J.Meunier, A. St-Arnaud, and J. Rousseau, "Multiple cameras fall dataset," *D.-U. d. Montréal*, Ed., ed, 2010.
[55] I. Charfi, J. Mitéran, J. Dubois, M. Atri, and R. Tourki, "Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification," *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 041106-041123, 2013, doi: 10.1117/1.JEI.22.4.041106.
[56] K. Adhikari, A. Bouchachia, and H. Nait-Charif, "Activity recognition for indoor fall detection using convolutional neural network," *15th IAPR International Conference on Machine Vision Applications(MVA)*, pp. 81-84, 2017, doi: 10.23919/MVA.2017.7986795.
[57] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer methods programs in biomedicine*, vol. 117, no. 3, pp. 489-501, 2014, doi: 10.1016/j.cmpb.2014.09.005.
[58] A. Shimanoe and M. Amemiya, "Identifying factors of public acceptance for usage of CCTV image," *Journal of the City Planning Institute of Japan*, vol.54, no. 3, pp. 750-757, 2019, doi: 10.11361/journalcpij.54.750.
[59] A. Zatserkovnyy and E. Nurminski, "Identification of Location and Camera Parameters for Public Live Streaming Web Cameras," *Mathematics*, vol. 10, no. 9, pp. 3601-3620, 2022, doi: 10.3390/math10193601.
[60] C.-B. Lin, Z. Dong, W.-K. Kuan, and Y.-F. J. A. S. Huang, "A Framework for Fall Detection Based on OpenPose Skeleton and LSTM/GRU Models," Applied Sciences, vol. 11, no. 1, pp. 329-348, 2020, doi: 10.3390/app11010329.
[61] L. Zhao and L. Wang, "A new lightweight network based on MobileNetV3," *KSII Transactions on Internet Information Systems*, vol. 16, no. 1, pp. 1-15, 2022, doi: 10.3837/tiis.2022.01.001.
[62] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 42, pp. 2011-2023, 2018, doi: 10.1109/TPAMI.2019.2913372.
[63] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ArXiv*, vol. abs/1704.04861, pp. 1-9, 2017, doi: 10.48550/arXiv.1704.04861.
[64] E. Kurniawan et al., "Deep neural network-based physical distancing monitoring system with tensorRT optimization," *International Journal of Advances in Intelligent Informatics*, vol. 8, no. 2, pp. 1-16, 2022, doi: DOI:10.26555/ijain.v8i2.824.
[65] L. Liu, E. B. Blancaflor, and M. B. Abisado, "A Lightweight Multi-Person Pose Estimation Scheme Based on Jetson Nano," *Applied Computer Science*, vol. 19, no. 1, pp. 1-14, 2023, doi: 10.35784/acs-2023-01.
[66] M. S. Pavithra, K. Saruladha, and K. Sathyabama, "GRU Based Deep Learning Model for Prognosis Prediction of Disease Progression," in *3rd International Conference on C*omputing Methodologies Communication, vol. 2019, pp. 840-844, 2019, doi: 10.1109/ICCMC.2019.8819830.
[67] A. Carlier, P. Peyramaure, K. Favre, and M. Pressigout, "Fall Detector Adapted to Nursing Home Needs through an Optical-Flow based CNN," *42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society*, vol. 2020, pp. 5741-5744, 2020, doi: 10.1109/EMBC44109.2020.9175844.
[68] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View Adaptive Neural Networks for High Performance Skeleton-Based Human Action Recognition," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 41, no. 8, pp. 1963-1978, 2019, doi: 10.1109/TPAMI.2019.2896631.
[69] I. Kareem, S. F. Ali, and A. Sheharyar, "Using Skeleton based Optimized Residual Neural Network Architecture of Deep Learning for Human Fall Detection," in *IEEE 23rd International Multitopic Conference*, vol. 2020, pp. 1-5, 2020, doi: 10.1109/INMIC50486.2020.9318061.
[70] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "Skeleton-based human activity recognition using ConvLSTM and guided feature learning," *Soft Computing*, vol. 26, no. 2, pp. 877-890, 2021, doi: 10.1007/s00500-021-06238-7.
[71] Y. Zheng, D. Zhang, L. Yang, and Z. Zhou, "Fall detection and recognition based on GCN and 2D Pose," *6th International Conference on Systems Informatics*, IEEE, vol. 2019, pp. 558-562, 2019, doi: 10.1109/ICSAI48974.2019.9010197.
[72] J. Lee and S. J. Kang, "Skeleton action recognition using Two-Stream Adaptive Graph Convolutional Networks," *36th International Technical Conference on Circuits/Systems, Computers Communications (ITC-CSCC)*. IEEE, vol. 2021, pp. 1-3, 2021, doi: 10.23919/CCC55666.2022.9901587.
[73] B. U. Maheswari, R. Sonia, M. P. Rajakumar, and J. Ramya, "Novel Machine Learning for Human Actions Classification Using Histogram of Oriented Gradients and Sparse Representation," Inf. Technol. Control., vol. 50, no. 4, pp. 686-705, 2021, doi: 10.5755/j01.itc.50.4.27845.
[74] S. Kiran *et al.*, "Multi-Layered Deep Learning Features Fusion for Human Action Recognition," *Computers Materials & Continua,* vol. 69, no. 3, pp. 1-15, 2021. doi: 10.32604/cmc.2021.017800.

Lei Liu

Lie Liu was born in Anhui, China, in 1987. He received his B.S. degree from the Anhui University, in 2010, the M.S. degree from Hefei University of Technology, in 2013, and PhD degree from the National University, Philippines, in 2023. He is currently a lecturer at the School of Computer Science, Huainan Normal University. His main research interests include computer vision and machine learning.

Yeguo Sun

Yeguo Sun was born in Anhui, China, in 1979. He received his B.S. degree from the Department of Mathematics and Computational Science, Fuyang Normal University, Fuyang, China, 2001, M.S. degree from the Department of Mathematic-s, Shanghai Normal University, Shanghai, China, 2007, and PhD degree from the Department of Control Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing, Chi-na, 2011. He is now a Professor at the School of Finance and Mathematics in Huainan Normal University. He has published several papers and some of them were indexed by SCI and EI. His current research interests include network control systems, neural networks, finite-time control.

Xianlei Ge

Xianlei Ge received his B.S. and M.S. degrees in information communication engineering from Chongqing University of Posts and Communications, Chongqing, China, in 2012 and 2015, respectively. From 2015 to 2016, he worked as a software engineer at Huawei Technologies Co., Ltd. He is a lecturer and researcher at Huainan Normal University since 2016. He is currently pursuing his Ph.D. degree in computer science at the College of Computing and Information Technologies, National University in Manila, Philippines. His research interests include image processing, machine learning and natural language processing.