Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 9962666, 15 pages
https://doi.org/10.1155/2022/9962666

Research Article

Lightweight Fall Detection Algorithm Based on AlphaPose Optimization Model and ST-GCN

Hongtao Zheng and Yan Liu

School of Information and Electrical Engineering, Zhejiang University City College, Hangzhou 310000, China

Correspondence should be addressed to Yan Liu; liuy@zucc.edu.cn

Received 9 May 2022; Accepted 9 June 2022; Published 11 July 2022

Academic Editor: Xuefeng Shao

Falls cause great harm to people, and the current, more mature fall detection algorithms cannot be well-migrated to the embedded platform because of the huge amount of calculation. Hence, they do not have a good application. A lightweight fall detection algorithm based on the AlphaPose optimization model and ST-GCN was proposed. Firstly, based on YOLOv4, the structure of GhostNet is used to replace the DSPDarknet53 backbone network of the YOLOv4 network structure, the path convergence network is converted into BiFPN (bidirectional feature pyramid network), and DSC (deep separable convolution) is used to replace the standard volume of spatial pyramid pool, BiFPN, and YOLO head network product. Then, the TensorRt acceleration engine is used to accelerate the improved and optimized YOLO algorithm. In addition, a new type of Mosaic data enhancement algorithm is used to enhance the pedestrian detection algorithm, improving the effect of training. Secondly, use the TensorRt acceleration engine to optimize attitude estimation AlphaPose model, speeding up the inference speed of the attitude joint points. Finally, the spatiotemporal graph convolution (ST-GCN) is applied to detect and recognize actions such as falls, which meets the effective fall in different scenarios. The experimental results show that, on the embedded platform Jeston nano, when the image resolution is 416 × 416, the detection frame rate of this method is stable at about 8.33. At the same time, the accuracy of the algorithm in this paper on the UR dataset and the Le2i dataset has reached 97.28% and 96.86%, respectively. The proposed method has good real-time performance and reliable accuracy. It can be applied in the embedded platform to detect the fall state of people in real time.

1. Introduction

Falls can cause all kinds of trauma, which can be life-threatening in severe cases. Studies also show that nearly half of all falls worldwide lead to medical attention, decreased functioning, impaired social or physical activity, and even death [1, 2]. Medical surveys have shown that if timely treatment can be performed after a fall, the risk of death can be reduced by 80% and the survival rate can be significantly improved. However, all actions taken after a fall are less important than detecting a person's posture before they fall. Therefore, it is of great significance to quickly detect the occurrence of falls [3].

At present, the research on fall detection can be divided into three main categories: (1) detection methods based on environmental equipment [4–6], which are detected according to the environmental noise formed when the human body falls, e.g., sensing the object's pressure and sound changes, are used to detect falls, however, this method has a higher false positive rate and is less likely to be adopted. (2) Detection methods based on wearable sensors [7–10], e.g., using accelerometers and gyroscopes, to detect falls, however, wearing sensors for a long time will affect people's comfort and increase the physical burden. The false positive rate is also higher for complex activities. (3) Detection methods based on visual recognition [11–15] can be divided into two categories: one is the traditional machine vision method to extract effective fall features. It requires low hardware requirements for the running platform, however, the robustness is not strong, and it is easily disturbed. The other type is artificial intelligence method, which uses the image captured by the image sensor for the training and reasoning of the convolutional neural network, and the recognition accuracy can reach a high level. However, at the same time, this method also requires a high training environment configuration, which greatly limits the application and promotion of this method. At the same time, in recent years, many embedded devices have appeared, such as Jeston nano, Jeston NX, Jeston TX2. Relatively cheap and small embedded devices also have considerable computing power, which provides the possibility for the migration and deployment of artificial intelligence algorithms. Most of the methods currently on the market cannot run well on embedded devices. Hence, this paper proposes a fall detection algorithm to solve this problem.

The specific improvement of the algorithm in this paper is as follows:

(1) In the early stage, to enhance the generalization ability of the dataset, the original mosaic data enhancement algorithm was improved and optimized, and a new mosaic data enhancement method was proposed.

(2) To reduce the structural complexity of the target detection algorithm, and at the same time, ensure a better recognition accuracy for people at different levels of complexity, this paper improves the structure of YOLOv4 and proposes a structure of a novel object detection algorithm.

(3) To improve the YOLO algorithm to a greater extent, this paper uses the TensorRt acceleration engine to accelerate.

(4) To ensure the accuracy of the detection algorithm, the joint detection algorithm selected in this paper is AlphaPose, and at the same time, considering the need to migrate AlphaPose to embedded devices, this paper proposes an optimization method for the detection model of AlphaPose.

(5) Introduce a spatiotemporal graph convolution algorithm as the actual detection of the fall state.

2. Related Work

At present, the most common and generally effective fall detection algorithm is the vision-based detection algorithm. Generally speaking, the overall operation logic of the vision-based detection algorithm is to first use the target detection algorithm to detect the pedestrians in the image and input the detection results into the joint point detection algorithms, such as AlphaPose and openpose, and finally according to the specific parameters of the joint points, the coordinates are combined with the behavioral state at the time of the fall to determine whether to fall.

2.1. Object Detection Algorithm Based on Pedestrian Detection. Traditional pedestrian detection methods mainly extract features manually. Tian et al. [16] propose a novel multiplex classifier model, which is composed of two multiplex cascades parts: Haar-like cascade classifier and shapelet cascade classifier. [17] proposed a histogram of oriented gradients (HOG), which exploits the directionality of edges to describe the overall appearance of pedestrians. However, the extraction steps of this extraction method are cumbersome, and the calculation of the recognition algorithm is complicated, resulting in poor real-time performance.

Pedestrian detection has achieved rapid progress because of recent developments in deep learning research. At present, target detection algorithms based on deep learning can be roughly divided into two categories: (1) two-stage detection algorithms represented by R-FCN (region-based fully convolutional neural network) [18] and (2) YOLO as the representative single-stage detection method (you only look once) [19]. The two-stage detection method has high accuracy and poor real-time performance. The single-stage detection method has slightly lower accuracy but has good real-time performance and fast detection speed.

The two-stage detection method realizes the cascade structure, the network calculation amount increases, and the accuracy is correspondingly improved, however, the detection speed is sacrificed accordingly, and the real-time requirements cannot be met. The problem has not been fixed well since then, although it has worked hard to make up for this shortcoming. Regarding the single-stage detection method, Redmon et al. proposed YOLO (you only look once) [19] in 2016, which is the first single-stage detection method based on deep learning. It creatively combines candidate regions with target recognition, which solves the problem of low efficiency of two-stage target detection algorithms. Redmon and Farhadi then went on to propose YOLOv2 [20] and YOLOv3 [21], which significantly improved the detection performance and enabled the YOLO family of methods to be widely used in various tasks. In 2020, Bochkovskiy improved the network structure of YOLOv3 and proposed YOLOv4. YOLOv4 greatly improves detection accuracy while ensuring speed. More recently, Jocher proposed YOLOv5, which brings together other state-of-the-art technologies. Compared with YOLOv4, although the performance of YOLOv5 is slightly worse, it is more flexible and faster than Yolov4 and has certain advantages in rapidly deploying models.

2.2. Development of Joint Detection Algorithms. In human pose detection, there are two main methods of joint point detection: bottom-up and top-down. The bottom-up approach is represented by Openpose [22], which is an end-to-end detection algorithm based on convolutional neural networks, supervised learning, and an open-source library developed with caffe as the framework. It can realize pose estimation, such as human motion, facial expression, movement, and so on. It has excellent robustness for single and multiplayer. The algorithm, firstly, detects all human body joint points in the image and then distinguishes which human body the joint points belong to through the relationship between the joint points. Although this method has a faster operation speed, it is easily disturbed by nonhuman bodies. The top-down method is represented by AlphaPose [23], which is a multistage detection method. Firstly, target detection is performed to identify the human target in the image and mark each human body area rectangle to exclude nonhuman interference, the detection of joint points for each human body area is very accurate, and the calculation speed is also fast.

2.3. Other Recommendations. Reference [24] proposed a multilayer dual LSTM network-based framework for multimodal sensor fusion to perceive and classify patterns of daily activities and highly shared events. Reference [25] proposed an optically anonymous image sensing system, which uses convolutional neural networks and autoencoders for feature extraction and classification to detect abnormal behaviors, which largely protects the privacy of the elderly. Reference [26] uses the two-dimensional image data to extract an effective image background through the frame difference method, Kalman filter, etc., and uses it as the input of KNN (K-nearest neighbor) classifier, which achieves an accuracy rate of 96%, and it is susceptible to variable factors. Reference [27] uses the two-dimensional image data to calculate optical flow information and sends it to VGG (visual geometry group) for feature extraction and classification of optical flow information to detect falls. In the literature [28], the feature information extracted by the CNN convolutional layer and the fully connected layer is sent to the long short-term memory (LSTM) network to train to extract the temporal correlation of human spatial actions and identify human behavior. LSTM needs to dynamically store and update data with limited real-time performance.

3. Materials and Methods

The basic flow of the fall detection algorithm in this paper is as follows: (1) regarding the training of the front weight file, the pedestrian dataset is collected by ordinary cameras and the new mosaic data enhancement method is used for data enhancement, and the target detection algorithm and the joint point detection algorithm are carried out, respectively. (2) Regarding the running process of the overall algorithm, the camera connected to Jeston nano captures real-time pedestrian images, uses the improved new YOLOv4 algorithm to accelerate the TensorRt engine to detect the target, and then converts the detection result to the tensor data structure to serialize the target image, invests in the Alpahpose joint point detection algorithm optimized by the model, and finally, the spatiotemporal graph convolutional neural network ST-GCN uses the coordinates of the key points of the human skeleton extracted by AlphaPose as the model input and constructs a joint as the graph node. The temporal relationship of the same joint is the spatiotemporal graph of the graph edge, taking the natural connection of human bones and the time relationship of the same joint as the time-space diagram of the edge of the graph, so that the information is integrated in the spatiotemporal and spatial domains. The final result is obtained by combining the motion analysis research. The specific algorithm structure flow chart is shown in Figure 1.

3.1. Object Detection Algorithm Based on Pedestrian Detection. The mosaic method was first proposed in the YOLOv4 paper. This method is based on the CutMix (cutting and mixing) [29] method to expand the generated data enhancement algorithm. The two blue paths in Figure 2 are mentioned in the YOLOv4 paper. m_1 represents the original image input, m_4 represents the image four-in-one input, and the innovation of the mosaic algorithm in this paper is that an input form m_9 is added under these two paths, which represents the image nine-in-one input. Once input, the specific generation flow chart is shown in Figure 3. Compared with m_4, m_9 greatly enriches the background of detected objects. In BN calculation, the data of 9 pictures can be calculated at a time, which makes the hardware resource requirements lower during training and can save more hardware resources.

The specific operation is as follows: the first step is to take the length and width (w, h) of the input image as a boundary value. Then, scale the image, where the x-axis and y-axis are, respectively, scaled to a certain multiple of k_x and k_y, whose formulas are as follows:

$$k_x = \text{Rand}\left(k_w, k_w + \Delta k_w\right), \tag{1}$$

$$k_y = \text{Rand}\left(k_h, k_h + \Delta k_h\right). \tag{2}$$

Among them, k_x and k_y are the minimum values of the length and width scaling multiples, respectively, and Δk_w and Δk_h are the lengths of the random size of the length-width scaling multiples, which are the hyperparameters. The Rand function is a random function.

The coordinates of the upper left corner and the lower right corner of the image after scaling are (A_i, B_i) and (a_i, b_i), and these four unknowns are obtained by the following formulas:

$$\begin{aligned} A_i &= \begin{cases} 0, & i = 1,2,3, \\ w \times k_1, & i = 4,5,6, \\ w \times k_2, & i = 7,8,9, \end{cases} \\ B_i &= \begin{cases} 0, & i = 1,4,7, \\ h \times k_3, & i = 2,5,8, \\ h \times k_4, & i = 3,6,9, \end{cases} \\ c_i &= A_i + w \times k_w, \\ d_i &= B_i + h \times k_h. \end{aligned} \tag{3}$$

Among them, k_1 and k_2 are the ratios of the distance between the upper left coordinate point and the 0 point of the two sets of images on the x-axis, except for the 0 point to the total width. Similarly, k3 and k4 are in the y-axis, except for the 0 point. k3 and k4 are the distance between the upper left coordinate point and the 0 point of the two sets of images and the total length ratio. The vertical dotted line in the figure is the picture width scale, accounting for one-tenth of the picture width, and the horizontal small dotted line is the picture length scale, accounting for one-tenth of the picture length. The first photo is of the same scale as the other eight photos, and the width and length are k_w and k_h times the original.

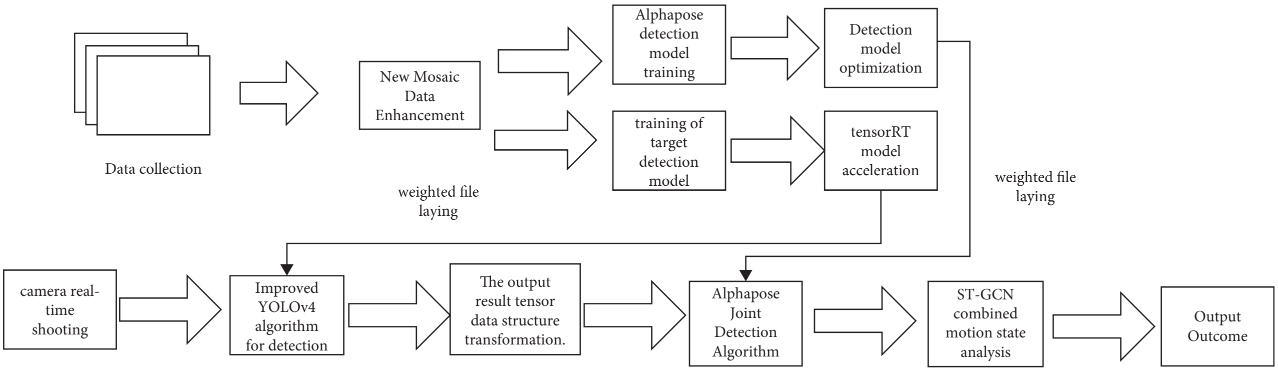
Data collection
New Mosaic Data Enhancement
Alphapose detection model training
training of target detection model
Detection model optimization
tensorRT model acceleration
weighted file laying
weighted file laying
camera real-time shooting
Improved YOLOv4 algorithm for detection
The output result tensor data structure transformation.
Alphapose Joint Detection Algorithm
ST-GCN combined motion state analysis
Output Outcome

Figure 1: Magnetization as a function of the applied field. *Note.* “Fig.” is abbreviated.

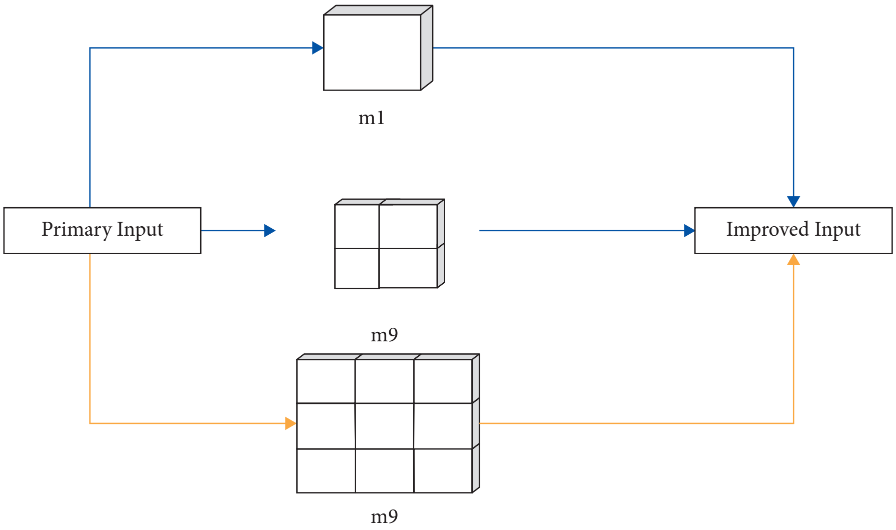

Figure 2: Improved mosaic data enhancement method.

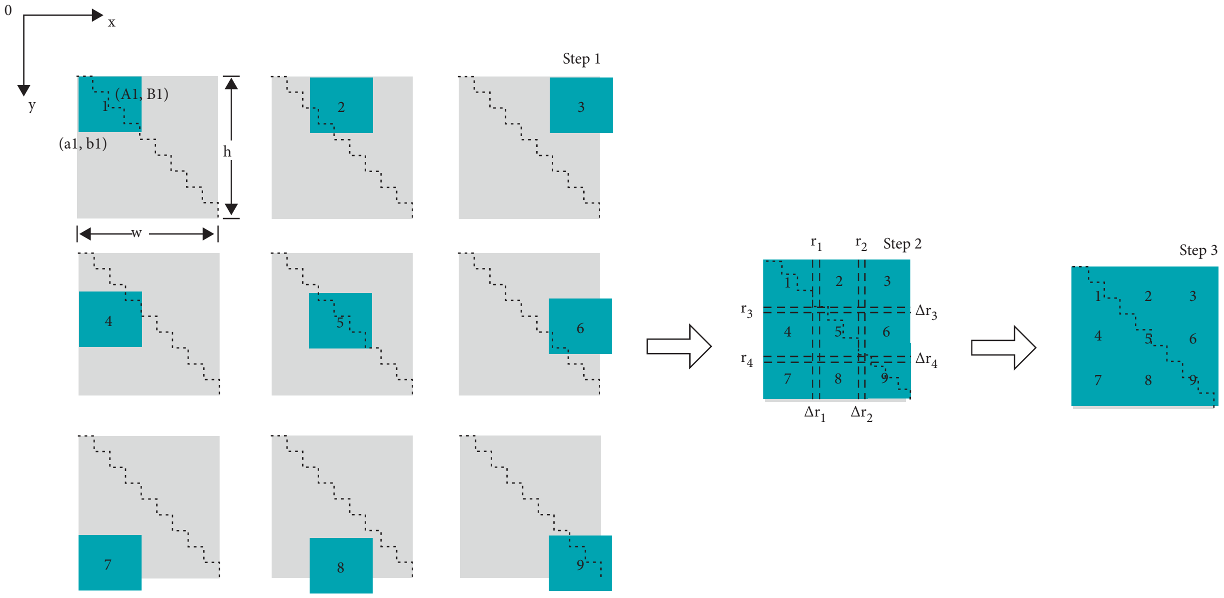

Figure 3: Mosaic nine-in-one data enhancement flowchart.

In step 2, flip, color gamut, and stitch the 9 photos cropped in the previous stage. Rely on the bounding box to limit the size of the stitched pictures, and crop the excess. There will be overlapping images. According to the schematic diagram of step 1 in Figure 3, the position of the small area needs to be reassigned, as shown in the following formula:

$$C_i' = \begin{cases} C_i, & C_1 < w, \\ w, & C_1 \geq w, \end{cases}$$
$$C_i' = \begin{cases} D_i, & D_1 < w, \\ h, & D_1 \geq w. \end{cases} \tag{4}$$

After the edge is cropped, use eight parallel dashed lines (as shown in step 2) to enclose four square areas, and use them as a random area for segmentation; k1, k2, k3, and k4 are the ratios of the coordinates of the segmentation line to the distance from the origin and the boundary. In the third stage, the inner overlapping part is to be cut for the second time, and the coordinate S_i of the dividing line can be obtained by the following formula:

$$S_i = \mathrm{Rand}\left(k_i, k_i + \Delta k_i\right) \quad i = 1, 2, 3, 4. \tag{5}$$

After cropping, the m_9 image stitching is completed. Since there will be some missing content in scaling and splicing, the edge targets of the original image may be cropped. Hence, the real boxes of these targets need to be cropped to meet the needs of target detection.

3.2. Structure Optimization of Human Object Detection Algorithm. The original AlphaPose human target detection algorithm uses YOLOv3, however, YOLOv4 proposed in recent years has significantly surpassed YOLOv3 in terms of detection accuracy and detection speed. It can cope with more complex detection environments (such as complex light and occlusion). However, because of the large amount of calculation, it is not suitable to migrate to embedded devices. Therefore, the human target detection algorithm in this paper is improved on the basis of the YOLOv4 algorithm structure, which ensures high pedestrian detection accuracy and faster recognition of frames.

The improvement of the specific structure is as follows: (1) the structure of GhostNet [29] is adopted to replace the DSPDarknet53 backbone network in the YOLOv4 network structure, which realizes the simplification of the network while maintaining the accuracy. (2) Convert the path aggregation network into BiFPN (bidirectional feature pyramid network) [30] to shorten the path from low-level information to high-level information and build the residual structure of the feature pyramid network to integrate richer semantic features and save spatial information. (3) DSC (deep separable convolution) [31] is adopted to replace the standard convolution of spatial pyramid pooling. BiFPN and YOLO head the network, which greatly reduces the amount of computation and improves network performance. The improved YOLOv4 algorithm structure is shown in Figure 4.

3.2.1. Human Feature Extraction Based on Ghostnet. Since the CSPDarknet53 structure in YOLOv4 requires a large amount of computation while efficiently extracting image features, this paper chooses a lightweight network structure like the GhostNet. The core idea of GhostNet is to use some operations with lower computational cost to generate the same features. There are many similarities between the network feature layers, and the redundant part in the feature layer may be an important part. Hence, GhostNet saves redundant information and obtains feature information with a lower computational cost.

The convolution block of GhostNet is the Ghost Module. Its function is to replace ordinary convolution. It divides ordinary convolution into two parts. Firstly, a 1 × 1 ordinary convolution is performed. For example, the convolution of 32 × 32 channels is normally used. But the GhostNet network uses 16-channel convolutions, the function of this 1 × 1 convolution is similar to feature integration, generating the feature concentration of the input feature layer. Then, we perform a depthwise separable convolution, which is a layer-by-layer convolution that uses the previous step to perform features. Condensation generates ghost feature maps.

The network structure combined with the GhostNet is shown in Figure 1, in which GBN is represented as GhostNetBottleNeck, which is a component of GhostNet. The GhostNetBottleNeck bottleneck layer consists of two GhostModules. The first is used to expand the number of channels, and the second is used to reduce the number of channels, matching the number of channels connected to the input. When the input is 416 × 416, the construction method of the GhostNet is shown in Table 1. When a picture is input into the GhostNet, we perform a 16-channel ordinary 1 × 1 convolution block (convolution + normalization + activation function). After that, the stacking of the ghost bottlenecks began. Using ghost bottlenecks, a 7 × 7 × 160 feature layer was finally obtained (when the input was 224 × 224 × 3). Then, a 1 × 1 convolution block is used to adjust the number of channels, and a 7 × 7 × 960 feature layer can be obtained at this time. After that, a global average pooling is performed, and then a 1 × 1 convolution block is used to adjust the number of channels to obtain a 1 × 1 × 1280 feature layer. Then, after tiling, the full connection can be performed for classification.

The operation of generating n feature images for any convolutional layer can be expressed as follows:

$$Y_0 = \mathrm{XOf} + b, \tag{6}$$

where $X \in R^{h\times c\times}w$, and $f \in R^{c\times k\times k\times m}$ is the convolution kernel of this layer. O represents the convolution operation, and b is the bias term. At this time, the feature map is as follows:

$$Y_0 \in R^{h'\times w'\times m'}. \tag{7}$$

The required floating-point number is $n \times h' \times w' \times c \times k \times k$. Assume that the ghost module contains an intrinsic feature map and $m \times (s - 1) = n/s \times (s - 1)$ linear transformation operations. The size of each operation kernel and the theoretical speedup of the ghost module upgrading the ordinary convolution are as follows:

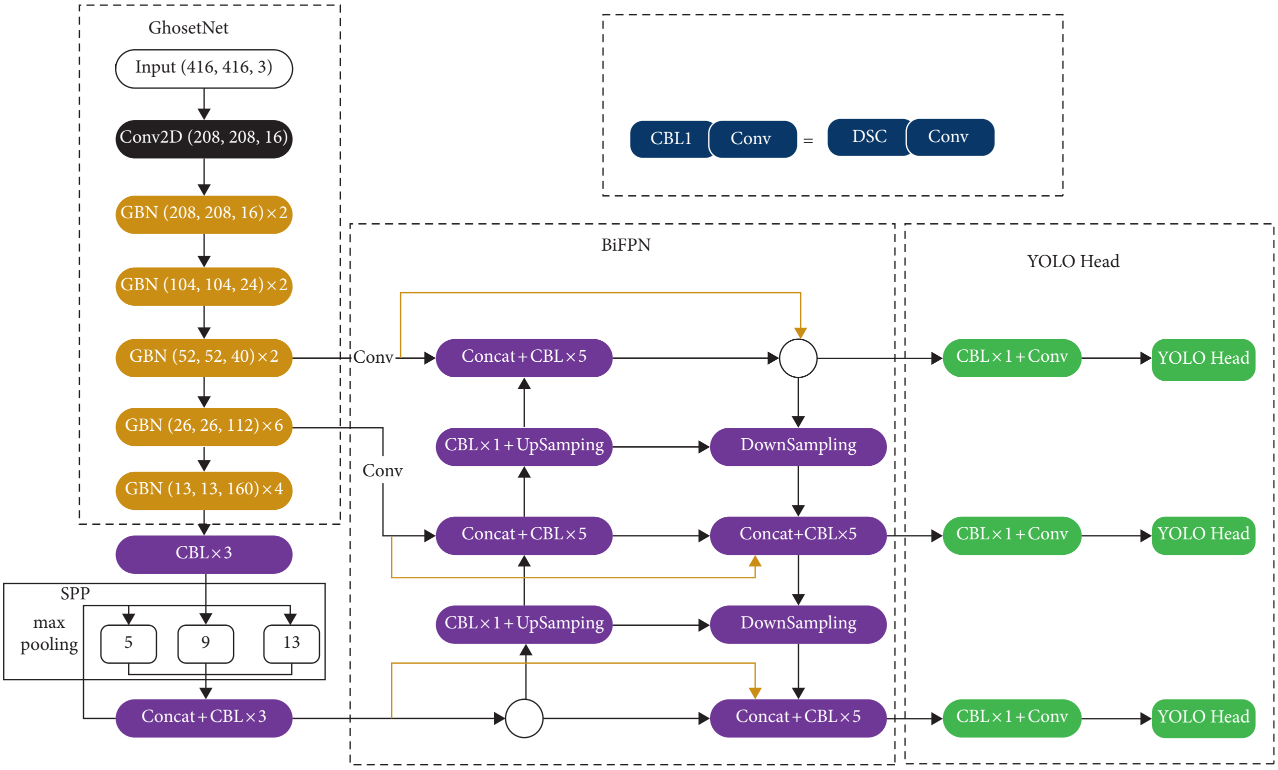

Figure 4: Improved structure of YOLOv4 algorithm.

Table 1: GhostNet construction method diagram.

Input	Operator	#exp	Out	SE	Stride
$416^2 \times 3$	Conv2d 3×3	—	16	—	2
$208^2 \times 16$	GBN	16	16	—	1
$208^2 \times 16$	GBN	48	24	—	2
$104^2 \times 24$	GBN	72	24	—	1
$104^2 \times 24$	GBN	72	40	1	2
$52^2 \times 40$	GBN	120	40	1	1
$52^2 \times 40$	GBN	240	80	—	2
$26^2 \times 80$	GBN	200	80	—	1
$26^2 \times 80$	GBN	184	80	—	1
$26^2 \times 80$	GBN	184	80	—	1
$26^2 \times 80$	GBN	480	112	1	1
$26^2 \times 112$	GBN	672	112	1	1
$26^2 \times 112$	GBN	672	160	1	2
$13^2 \times 160$	GBN	960	160	—	1
$13^2 \times 160$	GBN	960	160	1	1
$13^2 \times 160$	GBN	960	160	—	1
$13^2 \times 160$	GBN	960	160	1	1
$13^2 \times 160$	Conv2d 1×1	—	960	—	1
$13^2 \times 960$	AvgPool 7×7	—	—	—	—
$1^2 \times 960$	Conv2d 1×1	—	1280	—	1
$1^2 \times 1280$	FC	—	1000	—	—

$$\begin{aligned} r_c &= \frac{n \times h' \times w' \times c \times k \times k}{(n/s) \times h' \times w' \times c \times k \times k + (s-1) \times (n/s) \times h' \times w' \times d \times d} \\ &= \frac{c \times k \times k}{(1/s) \times c \times k \times k + ((s-1)/s) \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx s, \end{aligned} \tag{8}$$

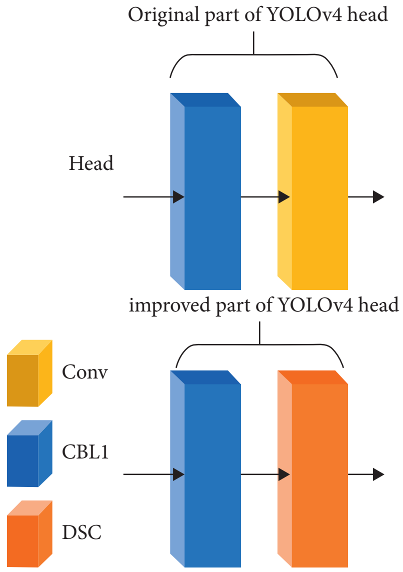

Figure 5: The modified part of CBL1 on the head of small-YOLOv4.

where $d \times d$ and $k \times k$ are similar. The theoretical parameter compression ratio is as follows:

$$r_c = \frac{n \times c \times k \times k}{n/s \times c \times k \times k + (s-1)/s \times n \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx s. \tag{9}$$

The theoretical parameter compression ratio of replacing ordinary convolution with the ghost module is approximately equal to the theoretical speedup ratio.

3.2.2. Improving Panet with reference to BIFPN. BiFPN (bidirectional feature pyramid network) was first proposed in the paper of EffientDet [31], and the author proposed that its purpose was to pursue a more efficient multiscale fusion method.

YOLOV4's original PANet adds a bottom-up channel based on FPN, and its CNN backbone provides a long path from the bottom to the top through more than 100 layers. In BiFPN, the input nodes and output nodes of the same layer can be connected across layers to ensure that more features are incorporated without increasing the loss. This algorithm performs cross-layer connections on the same level of PANet (the three orange lines in Figure 4). In this way, the path from low-level information to high-level information can be shortened, and their semantic features can be combined together. In BiFPN, adjacent layers can be merged in series. In this paper, the adjacent layers of PANet are merged in series (the two blue lines in Figure 4).

The improved PANet has the characteristics of bidirectional cross-scale connection and weighted feature fusion, which improves the feature fusion ability and further increases the feature extraction ability.

3.2.3. DSC Replaces Standard Convolution. In the algorithm of this paper, the 1×1 standard convolutional network in the CBL1 module of the YOLOv4 head is replaced with DSC (deep separable convolution), which further reduces the network computing cost in practical applications. The modified part of CBL1 is shown in Figure 5. The standard convolutional network calculation uses a weight matrix to realize the joint mapping of spatial dimension features and channel dimension features at the cost of high computational complexity, high memory overhead, and many weight coefficients.

DSC specifically divides the traditional convolution operation into two steps. Assuming that the original convolution is 3×3, DSC is to first convolve M feature maps of M 3×3 convolution kernels one-to-one. M results are generated directly without summing. Then, the M results previously generated are normally convolved with N 1×1 convolution kernels, summed, and finally, N results are generated. Therefore, the literature [17] divides DSC into two steps, as shown in Figure 6 below. One step is called depthwise convolution, which is B in the figure below, and the other step is pointwise convolution, which is C in the figure below.

Assuming that the size of our input feature map is $D_F \times D_F$, the dimension is M, the size of the filter is $D_k \times D_k$, the dimension is N, and assuming that the padding is 1, the stride is 1. Hence, the original convolution operation requires the following number of matrix operations: $D_k \times D_k \times M \times N \times D_F \times D_F$. The parameter of the convolution kernel is $D_k \times D_k \times M \times N$, and the number of matrix operations that DSC needs to perform is $D_k \times D_k \times M \times D_F \times D_F + M \times N \times D_F \times D_F$. The parameter of the convolution kernel is $D_k \times D_k \times M + N \times M$. Since the convolution process is mainly a process of reducing spatial dimension and increasing channel dimensions, namely $N > M$, the convolution kernel parameter of standard convolution is larger than that of DSC. At the same time, the ratio of the parameter quantity of DSC to the standard convolution parameter quantity is as shown in equation (4).

From equation (4), we can get a convolution kernel with a size of 3×3, which reduces the computation to 11.1% of the standard convolution.

3.3. Structure Optimization of Human Object Detection Algorithm. Commonly used model compression methods are as follows: network pruning, knowledge distillation, model quantization, etc. Since the network structure used in this paper is replaced by the lightweight GhostNet network, if the network continues to be pruned, it is very likely to destroy the integrity of the model and have a greater impact on the accuracy. Therefore, this paper uses model quantization to further reduce the number of parameters and model size.

The quantization method is further divided into quantization-aware training and post-training quantization. The post-training quantization method is divided into hybrid quantization, 8-bit integer quantization, and half-precision floating-point quantization. Post-training quantization directly quantizes the model after ordinary training. The process is simple, and there is no need to consider the quantization problem during the training process. The

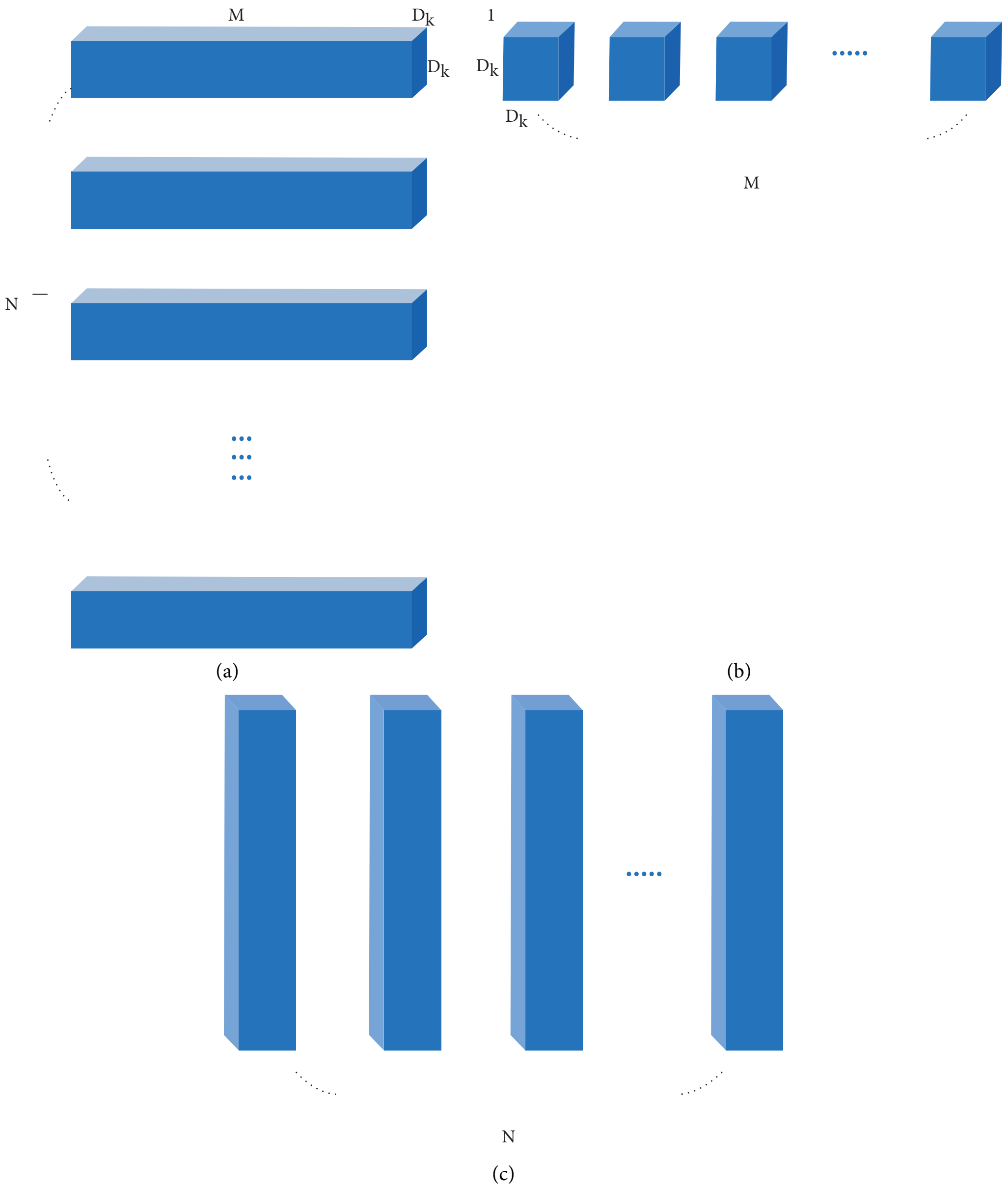

Figure 6: Structure diagram of DSC. (a) Stand convolution filters. (b) Depthwise convolution filters. (c) Depthwise separable convolution.

accuracy of the model with a large amount of parameter redundancy is lost.

This paper uses the TensorRT acceleration engine to convert the model weight file into an int8 type trt file using the post-training quantization method and performs overall optimization through a series of operations, such as tensor fusion, kernel adjustment, and multistream execution. Figure 7 is a schematic diagram of the overall optimization of TensorRT.

3.4. Structure Optimization of Human Object Detection Algorithm.

After the detection result is obtained through the target detection algorithm, the detection result is converted into a 2-dimensional tensor data structure, and the specific data structure form is shown in equation (9).

$$T_d = [[x_1, y_1, w_1, h_1, c_1], [x_2, y_2, w_2, h_2, c_2], \ldots, [x_i, y_i, w_i, h_i, c_i]], \quad (10)$$

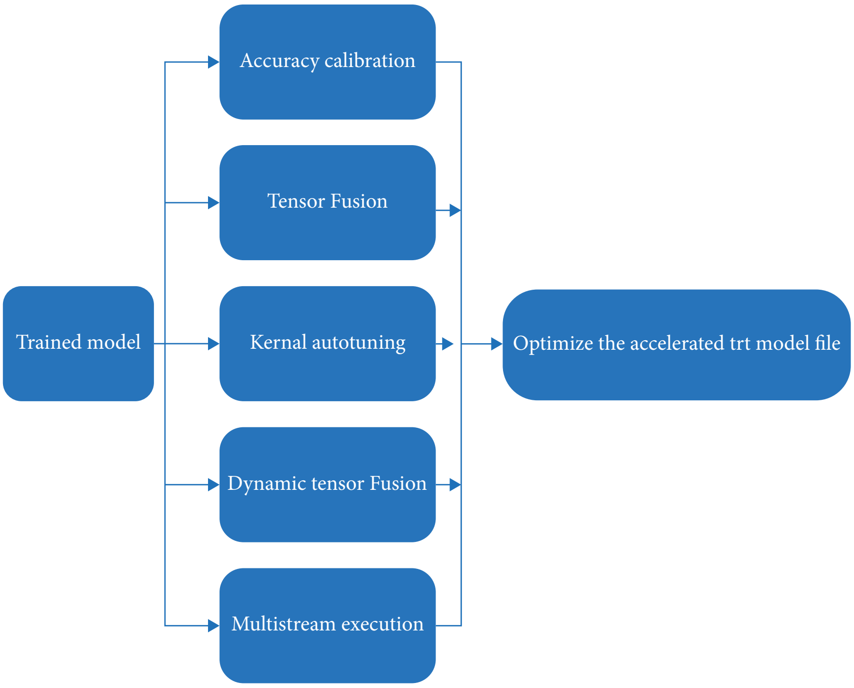

FIGURE 7: Schematic diagram of TensorRT optimization.

where $[x_i, y_i, w_i, h_i, c_i]$ represents the structured data of the i^{th} pedestrian, and x represents the upper left corner of the prediction box.

The original image T_m is transformed into a floating-point 32-bit tensor type data T_t. Hence, formula (1) represents a normalization operation on Im_t, whereT_t [0] is the R channel data of Im, G channel data of T_t [1], and B channel data for T_t [2].

$$\begin{cases} T_t[0] + = -0.416, \\ T_t[1] + = -0.461, \\ T_t[2] + = -0.479. \end{cases} \tag{11}$$

According to T_d, the human body area images are cut out from the original images, and they are arranged in the descending order of confidence to obtain a serialized image list, which realizes the serialization of human body images and improved data interaction efficiency between the target detection model and the human joint point detection model.

3.5. Optimization of Algorithm Model for Pose Joint Point Detection. The algorithm of AlphaPose in the original text uses the Fast_Reset50-based network, and the optimization method is shown in Figure 8.

The pose joint point detection model inputs dummy network layer dimension initialization, and the dummy network layer input dimension is set to tensor type $(1,3,H_{\text{dummy}}, W_{\text{dummy}})$, where 1 means that the batchsize is 1, 3 means the number of image channels, and W_{dummy}, H_{dummy} indicates the network layer input image normalization scale. In this paper, $W_{\text{dummy}} = 160$ and $H_{\text{dummy}} = 224$. Customize the design for the input and output network layers of the dimensionally initialized model. The input layer is set to input, and the output layer is set to output. Create a target detection model calculation graph, set the input dimension of the calculation graph to $(1, 3, W_d, H_d)$, where 1 means the batchsize is 1, 3 means the number of image channels, and W_d, H_d means the network layer input image normalization scale. $W_d = 160$, $H_d = 224$ in this paper. Load the model conversion optimizer to generate the pose joint detection optimization model AlphaPose-trt.

3.6. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. Using the spatio-temporal graph convolutional network ST-GCN [32], using the coordinates of human skeleton key points output by the AlphaPose algorithm as model input, construct a graph node with joint points as the natural connection of the human skeleton and the same joints. The temporal relationship is a spatiotemporal graph of graph edges, so that information is integrated in the temporal and spatial domains.

The spatiotemporal graph convolutional neural network is divided into a spatial graph convolution and temporal graph convolution. Spatial graph convolution is to construct spatial graph convolution within frames based on the natural connectivity of human joints. Spatial graph convolution is to construct spatial graph convolution within the frame according to the natural connectivity of human joint points, which can be recorded as $G_S = (V_S, E_S)$, where $V_S = \{v_{ti} | i = 1,2, \ldots, N_S\}$ represents all the joint points in a skeleton, and $E_s \left\{ v_{ij} v_{ij} / (i, j) \in H \right\}$ represents the connection between the joint points. Each node is described by a feature vector $F(V_i)$ to describe the spatial feature, which is represented by the spatial graph convolution which is obtained. Temporal graph convolution connects the same nodes in consecutive multiframe images on the spatial graph to form

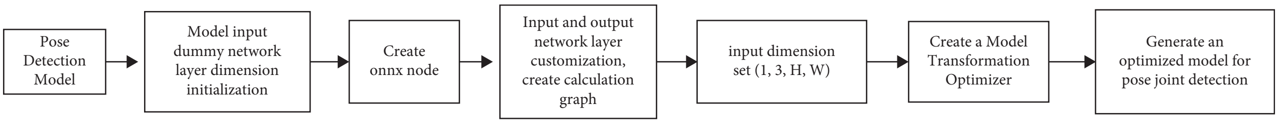
Pose Detection Model
Model input dummy network layer dimension initialization
Create onnx node
Input and output network layer customization, create calculation graph
input dimension set (1, 3, H, W)
Create a Model Transformation Optimizer
Generate an optimized model for pose joint detection

Figure 8: Optimization steps of AlphaPose detection model.

the spatial-temporal graph of the skeleton sequence, denoted as $G_T = (V_T, E_T)$. $V_T = \{v_{ti} | t = 1,2, \ldots, N_t\}$ represents the joint point sequence of the same part, and $E_T = \left\{v_{ij} v_{(t+l)}\right\}$ represents the connection between them, as shown in Figure 9.

The spatiotemporal graph convolution algorithm combines the motion analysis research to divide the spatial graph into three subsets, which represent the features of centripetal motion, eccentric motion, and rest, respectively. The root node is the selected skeleton joint point itself, including static features. Connecting the neighbor nodes closer to the center of gravity of the skeleton than the root node includes centripetal motion features. Connecting the neighbor nodes farther from the root node than the center of gravity of the skeleton includes centrifugal motion features. The three subset convolution results express action features at different scales, respectively.

The spatiotemporal graph convolutional neural network model takes the joint coordinate vector of the graph node as input and extracts deeper features through the 9-layer ST-GCN convolution module. The feature dimension of each node is 256, and the key frame dimension is 38. Then, the obtained tensors are globally pooled, and backpropagation is used to train the model end-to-end. Finally, the SoftMax classifier obtains the corresponding action category probability and outputs the action with the highest probability. Each ST-GCN layer adopts the Resnet structure to enhance the gradient propagation and adds a dropout strategy to the ST-GCN layer to solve the gradient explosion problem. The overall flow of the model is shown in Figure 10.

4. Experiments and Analysis

4.1. Dataset Analysis. The datasets used for training in this experiment mainly include 20 categories of VOC2007 and VOC2012, and 10,000 datasets of people that the author randomly collected. Through the program, VOC2012 and VOC2007 only retain the label information of this category. The dataset of 10,000 people collected by the author is divided into the training set, validation set, and test set according to the ratio of 6 : 2 : 2. The final number of images is shown in Table 2.

4.2. Anchor Box. To be more suitable for the category of person, the prior frame in the improved target detection algorithm in this paper is obtained by the K-means clustering dataset method. The image input in this paper adopts 416 × 416, and the clustering iteration reaches 73 times. The union ratio of the box and the prior box reaches 78.91%, and nine a priori boxes are obtained, as shown in Table 3.

4.3. Training and Operation Environment. The model training platform in our laboratory is RTX 3090, video memory 24G, etc. The specific parameters are shown in Table 4. The network model is trained on the deep learning framework of Tensorflow2.5 based on GhostNet and CSPDarknet53. All input images are of size 416 × 416. The follow-up effect verification and testing platform of the experiment is with Jeston nano.

4.4. Evaluation Criteria. We use FPS, precision, mAP, accuracy, *F*-score, sensitivity, specificity, and other indicators to evaluate our proposed method. The test set is divided into two categories, one is positive samples and the other is negative samples. TP is the number of positive samples predicted as positive samples. FP is the number of negative samples predicted as positive samples. FN is the number of predicted positive samples as negative samples. TN is the number of predicted negative samples as negative.

4.4.1. FPS (Frames per Second). The evaluation standard of detection speed used in this paper is FPS, which refers to the number of frames per second. The larger the FPS, the more frame rates the American Standard transmits, and the smoother the displayed image. To meet the real-time requirements of human body detection, the larger the FPS value, the smoother the picture seen, and the better the effect.

4.4.2. mAP (Mean Average Precision). The definition of the mAP is shown in equation (12), which represents the average value of the average precision APi of n types of targets, and $n = 1$ in this experiment.

$$\text{mAP} = \frac{\sum AP}{N(\text{Class})} = \sum AP. \tag{12}$$

4.4.3. Accuracy. Accuracy is a commonly used evaluation index. Generally speaking, the higher the accuracy rate, the better the classifier.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{13}$$

4.4.4. Precision. Precision can measure the accuracy of object detection, specifically defined as shown in equation (14) below.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{14}$$

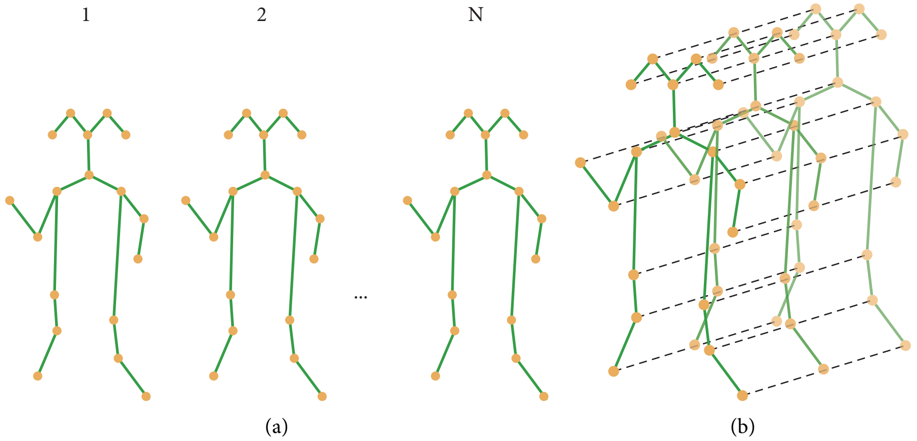

Figure 9: Construction of the spatio-temporal map of human joint points. (a) Bone space map sequence of N frames. (b) Skeleton space time-diagram (arrows indicate time series edges).

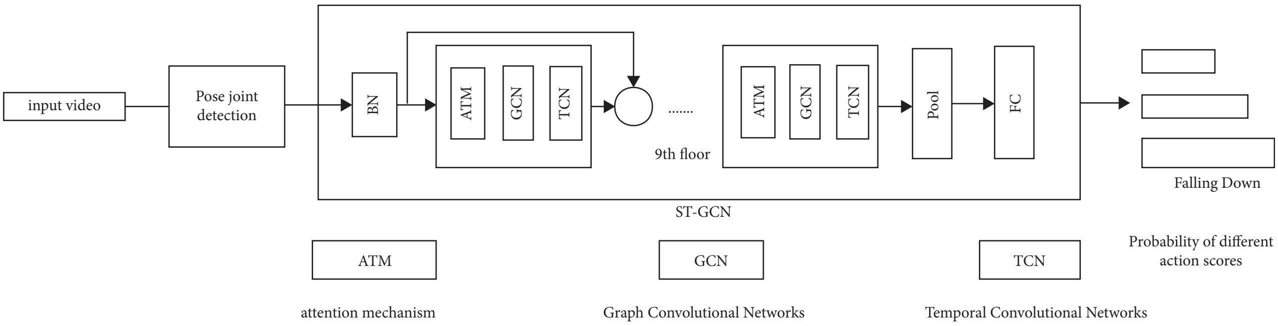

Figure 10: The overall framework of ST-GCN.

Table 2: The number of various data sets.

Types of data sets	Total number
Training sets	14265
Test sets	4765
Validation set	4755

Table 3: A priori frame size.

Size	Anchor box
13 × 13	(234, 280), (260, 379), (377, 354)
26 × 26	(107, 270), (135, 351), (171, 190)
52 × 52	(27, 44), (60, 121), (75, 230)

Table 4: Software and hardware configuration.

Component	Configuration
Operating system	Ubuntu 18.04
Memory	64
GPU	Nvidia GeForce RTX 3090
GPU acceleration library	CUDA 11.2 cuDNN v8.2.1
Deep learning framework	Tensorflow2.5
Programming language	Python3.9

4.4.5. F-Score. The *F*-score indicator combines the results of precision and recall outputs. The value of *F*-Score ranges from 0 to 1, where 1 represents the best output result of the model, which is specifically defined as shown in equation (15) below.

$$F - \text{score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{15}$$

4.4.6. Sensitivity. Sensitivity represents the sensitivity, which represents the predictive ability of positive examples (the higher, the better), and it is numerically equal to the recall rate, which is specifically defined as shown in equation (16) below.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{16}$$

4.4.7. Specificity. Sensitivity represents the predictive power of positive examples (higher is better), and the specific definition is shown in equation (17) below.

Table 5: The influence of mosaic data enhancement method on target recognition accuracy under different proportions.

Algorithm type	$m_1:m_4:m_9$	Dim light (%)	Chaotic environment mAP (%)	Human body occlusion mAP (%)
Algorithm in this paper	1 : 0 : 0	72.28	71.54	75.55
	0 : 1 : 0	60.11	65.58	66.18
	0 : 0 : 1	41.74	50.69	50.10
	1 : 1 : 0	72.45	71.99	75.10
	1 : 0 : 1	70.71	70.24	73.56
	1 : 1 : 1	72.19	71.58	76.01
	2 : 2 : 1	**76.28**	**76.68**	**78.10**
	2 : 1 : 1	72.27	70.44	73.56
	3 : 2 : 1	72.78	72.57	76.15
	4 : 2 : 1	73.35	72.16	74.20
	4 : 3 : 2	72.80	72.01	75.98
	5 : 3 : 2	74.01	74.48	77.52
YOLOv4	1 : 1 : 0	75.90	75.56	76.14

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{17}$$

4.5. Evaluation Criteria

4.5.1. A Novel Mosaic Data Augmentation Method. The new Mosaic data enhancement method in this paper is used to enhance the dataset, and the image input ratio of the three paths of m_1, m_4, and m_9 in Figure 2 that can maximize the accuracy of identifying complex situations is a problem that needs to be discussed. Table 5 below shows the influence of different input ratios of m_1, m_4, and m_9 on the accuracy of human recognition in three complex situations of dim light, chaotic environment, and human occlusion in the dataset. It can be seen from this table that when m_1, m_4, and m_9 ratio is 2 : 2 : 1, the effect of data enhancement is most obvious.

4.5.2. Target Detection Algorithm Network Improvement Effectiveness. To verify the impact of the improvement of the target detection algorithm on the performance of the YOLOv4 model, the above three improved methods were designed for ablation experiments on Jeston nano for more adequate comparison, thus proving the necessity and effectiveness of the proposed method. Among them, "+" indicates that the improved method is used in the experiment, "–" indicates that the method is not used, and the test indicators in this table refer to the detection effect of the human body in the test set of this paper. As can be seen from Table 6, after replacing the backbone network with the GhostNet, although the mAP value for the identification of person categories has been slightly reduced, the running frame rate has been significantly improved. After the introduction of BiFPN, the running frame rate has basically not changed, however, the mAP value has been greatly improved. Using the depthwise separable convolution to replace the ordinary convolution in the original YOLOv4 head, the running frame rate is significantly improved while the mAP value is slightly reduced. Compared with YOLOv4, the improved network structure has a slight decrease in the mAP value for the detection effect of Person, however, at the same time, the running frame rate has been significantly improved, which meets the basic ability of running on embedded devices. Finally, we chose to use the TensorRt framework to accelerate, and after using the TensorRt framework, the runnable frame rate was greatly improved, while the mAP value remained basically unchanged.

4.5.3. Comparison of Optimization Effectiveness of AlphaPose Algorithm Model. To verify the effectiveness of the AlphaPose algorithm model optimization method in this paper, this paper chooses to compare the effects of three models, including openpose, AlphaPose, and AlphaPose-trt. The mAP value in this paper is the human detection effect for the test set of this paper. The results of running on Jeston nano are shown in Table 7 below. It can be seen from Table 7 that the frame rate of openpose is lower than that of AlphaPose, while the mAP value is also lower than that of AlphaPose. Compared with the original model (AlphaPose), the optimized model (AlphaPose-trt) has a stable mAP value and greatly improves the running frame rate.

4.5.4. Comparison of Effectiveness of Fall Detection Algorithms. Because of the need to further demonstrate the overall advantages of the algorithm in this paper in detection accuracy and running frame rate, we need to compare the algorithm in this paper with other computer vision algorithms of the same type, however, considering that many of the more popular algorithms are not open source, it is impossible to migrate to Jeston nano to run. Hence, the selected comparison algorithms cannot have an accurate running frame rate, however, after analyzing the structure of these algorithms, it can be concluded that these algorithms are computationally complex and require a large number of calculations, and they do not have the ability to migrate to embedded devices. The final results are shown in Table 8. The data in this table is analyzed, and various evaluation data for human fall detection are tested in the Le2i fall and UR fall datasets, respectively. Compared with this paper, the literature [33] has achieved better results. The reason for the F1-score is because they employ a two-pass ensemble, using two classifiers, including random forest (RF) and multilayer perceptron (MLP), to identify falls, however, it leads to more

Table 6: Ablation study on the people dataset.

	GhostNet	Bi-FPN	DSC	TensorRt	mAP (%)	FPS
YOLOv4	–	–	–	—	87.27	2.35
	+	–	–	—	86.38	6.32
	–	+	–	—	87.40	2.23
	–	–	+	—	86.51	3.12
	+	+	–	—	87.08	6.13
	+	+	+	—	86.81	8.02
Our method	+	+	+	+	86.81	**24.33**

Table 7: The report posture detection model performance comparisons.

Pose estimation model	Frame rate	mAP (%)	Resolution of the image
Openpose	3.66	71.11	416 × 416
AlphaPose	7.72	82.12	416 × 416
AlphaPose-trt	13.13	82.05	416 × 416

Table 8: Comparison of different fall detection algorithms.

Algorithm type	Dataset	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	*F*-score	FPS
Wang et al. [33]	Le2i fall	96.91	97.65	96.51	97.37	97.08	—
Chamle et al. [35]	Le2i fall	79.31	79.41	83.47	73.07	81.39	
Our method	Le2i fall	96.86	97.01	96.71	96.81	96.77	8.33
Wang et al. [33]	UR fall	97.33	97.78	97.78	96.67	97.78	—
Harrou et al. [34]	UR fall	96.66	94	100	94.93	96.91	—
Our method	UR fall	97.28	97.15	97.43	97.30	97.29	8.33

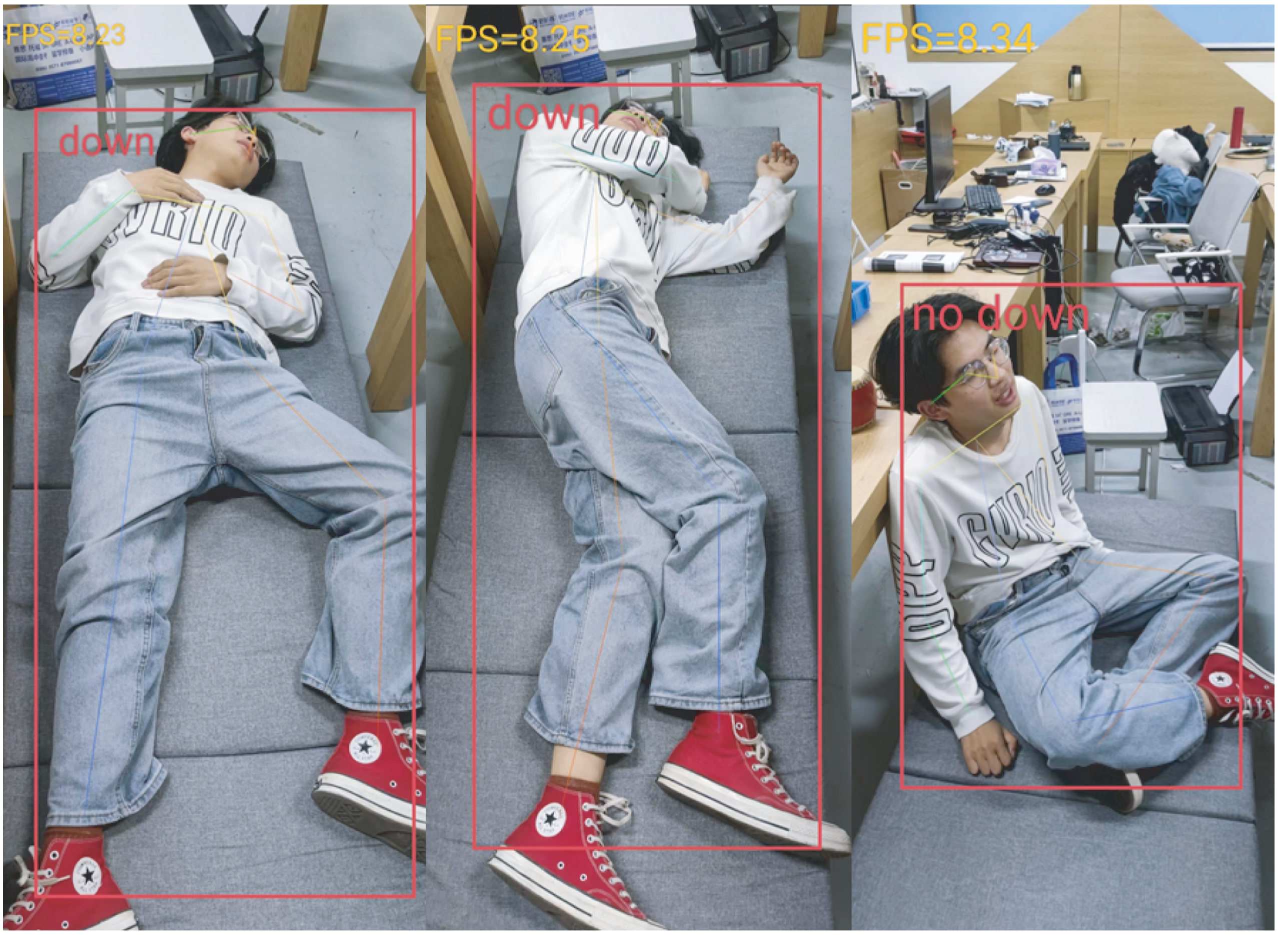

Figure 11: Effect diagram of experimental results.

computational complexity. It may also take more time from the classifier to the ensemble result, which leads to the poor real-time and transferability of the detection method. In contrast, the F1-score of the algorithm in this paper is slightly lower than that of [33]. At the same time, the real-time performance and migration are excellent. Compared with the methods of [34, 35] under the same dataset, the algorithm in this paper also has advantages in migration and real-time performance, and it also achieves a better balance in the two indicators of sensitivity and specificity. The results of analyzing the two validation datasets are similar, which further proves the stability of the algorithm in this paper. Figure 11 shows the detection results of the fall detection algorithm in this paper.

5. Conclusions and Future Work

This paper mainly studies the fall detection method based on computer vision technology. This method combines YOLO, AlphaPose, and ST-GCN. Through YOLO and AlphaPose, the key points and position information of the human body are obtained then output the recognition result through the spatiotemporal graph convolutional network. ST-GCN takes the output coordinates of the key points of human skeleton as a model input and constructs a spatiotemporal graph with joint points as graph nodes, natural connections of human skeletons, and the temporal relationship of the same joints as graph edges, so that the information is in the time and space domains that are integrated together.

The experimental results show that the method is transferable. In this paper, the improvement and optimization of the YOLOv4 algorithm and the effectiveness of the detection model optimization of AlphaPose are obtained under the running test of VOC07 + 12 and the self-made dataset. In addition, through the more popular fall detection algorithm in recent years and the test and verification of the algorithm in this paper in the UR Fall dataset, it is concluded that the algorithm in this paper has a high running frame rate on the basis of the detection accuracy, which is not much different from other algorithms, and it has better mobility and better adaptability in embedded devices.

In the future, we will focus more on complex fall detection and multiperson detection, such as outdoor fall detection and crowd trampling. At the same time, combined with the high applicability of embedded devices, we will integrate algorithms into real life, such as fall detection algorithms and monitoring systems. At the same time, there are many details that need to be improved for the operation effect of the algorithm in this paper, and we will continue to work hard.

Data Availability

The datasets used to support the findings of this study are available from the authors upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work received support from the Industry University Research Innovation Fund of Science and Technology Development Center of the Ministry of Education (No. 2021JQR004), Public Welfare Projects in Zhejiang Province (No. LGF20F030002), Project of Hangzhou Science and Technology Bureau (No. 20201203B96), 2021 National Innovation Training Project for College Students (No. 202113021008), The Ministry of Education Industry-University Cooperation Collaborative Education Project (202102019039), and Zhejiang University City College Scientific Research Cultivation Fund Project (J-202223). It is supported by the Zhejiang University Student Science and Technology Innovation Activity Plan (Xinmiao Talent Plan), project number: 2021R437010.

References

[1] J. Gutiérrez, V. Rodríguez, and S. Martin, “Comprehensive review of vision-based fall detection systems,” *Sensors*, vol. 21, no. 3, p. 947, 2021.

[2] M. Mubashir, L. Shao, and L. Seed, “A survey on fall detection: principles and approaches,” *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[3] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, “A high reliability wearable device for elderly fall detection,” *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4544–4553, 2015.

[4] M. Alwan, P. J. Rajendran, and S. Kell, “A smart and passive floor-vibration based fall detector for elderly,” in *Proceedings of the Information & Communication Technologies*, Ictta. IEEE, Berkeley, CA, USA, May 2006.

[5] Y. Zigel, D. Litvak, and I. Gannot, “A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, 2009.

[6] S. M. Khan, M. Yu, P. Feng, L. Wang, and J. Chambers, “An unsupervised acoustic fall detection system using source separation for sound interference suppression,” *Signal Processing: The Official Publication of the European Association for Signal Processing (EURASIP)*, vol. 110, 2015.

[7] J.-S. Lee and H.-H. Tseng, “Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers,” *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8293–8302, 2019.

[8] X. Xi, W. Jiang, and L. ü Zhong, “Daily activity monitoring and fall detection based on surface electromyography and plantar pressure,” *Complexity*, vol. 2020, Article ID 9532067, 12 pages, 2020.

[9] O. Kerdjidj, N. Ramzan, and K. Ghanem, “Fall detection and human activity classification using wearable sensors and compressed sensing,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, 2019.

[10] S. Angela and J. José Vargas-Bonilla, “Real-life/real-time elderly fall detection with a triaxial accelerometer[J],” *Sensors*, vol. 18, no. 4, p. 1101, 2018.

[11] S. G. Miaou, P. H. Sung, and C. Y. Huang, “A customized human fall detection system using omni-camera images and personal information,” in *Proceedings of the Conference on Distributed Diagnosis & Home Healthcare*, IEEE, Arlington, Virginia, April 2006.

[12] F. Merrouche and N. Baha, "Depth camera based fall detection using human shape and movement," in *Proceedings of the IEEE International Conference on Signal & Image Processing*, IEEE, Beijing, China, September 2017.

[13] K. H. Chen, Y. W. Hsu, and J. J. Yang, "Enhanced characterization of an accelerometer-based fall detection algorithm using a repository," *Instrumentation Science & Technology: Designs and applications for chemistry, biotechnology, and environmental science*, vol. 45, 2017.

[14] W. N. Lie, A. T. Le, and G. H. Lin, "Human fall-down event detection based on 2D skeletons and deep learning approach," in *Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT)*, IEEE, Chiang Mai, Thailand, January 2018.

[15] A. Lotfi, S. Albaw endi, H. Powell, K. Appiah, and C. Langensiepen, "Supporting independent living for older adults; employing a visual based fall detection through analysing the motion and shape of the human body," *IEEE Access*, vol. 6, pp. 70272–70282, 2018.

[16] H. Tian, Z. Duan, and A. Abraham, "A novel multiplex cascade classifier for pedestrian detection," *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1687–1693, 2013.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for hu-man detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[18] J. Dai, Y. Li, and K. He, "Object detection via region-based fully convolutional networks," *Advances in Neural Information Processing System*, vol. 29, 2016.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "Youonlylookonce:unified,realtimeobjectdetection," in *Proceedings of the 2016IEEEConferenceonComputerVisionandPatternRecognition*, pp. 779–788, IEEE, Las Vegas, NV, USA, June 2016.

[20] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pat-tern recognition*, pp. 7263–7271, Honolulu, HW, USA, July 2017.

[21] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[22] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: regional multi-person pose estimation," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2353–2362, Venice, Italy, October 2017.

[23] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using Part Affinity fields," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1302–1310, Honolulu, HI, USA, July 2017.

[24] H. Li, A. Shrestha, and H. Heidari, "Bi-LSTM network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1191–1201, 2019.

[25] C. Ma, A. Shimada, H. Uchiyama, H. Nagahara, and R.-i. Taniguchi, "Fall detection using optical level anonymous image sensing system," *Optics & Laser Technology*, vol. 110, pp. 44–61, 2019.

[26] K. De Miguel, A. Brunete, M. Hernando, and E. Gambao, "Home camera-based fall detection system for the elderly," *Sensors*, vol. 17, no. 12, p. 2864, 2017.

[27] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras, "Vision-based fall detection with convolutional neural networks," *Wireless Communications and Mobile Computing*, vol. 2017, no. 1, pp. 1–16, 2017.

[28] H. Gammulle, S. Denman, and S. Sridharan, "Two stream LSTM: a deep fusion framework for human action recognition," in *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision*, pp. 24–31, Piscataway:IEEE, Honolulu, HI, USA, July 2017.

[29] Y. Yang, G. Xie, and Y. Qu, "Real-time detection of aircraft objects in remote sensing images based on improved YOLOv4," in *Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 1156–1164, Chongqing, China, March 2021.

[30] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: scalable and efficient object detection," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, Seattle, Washington, USA, June 2020.

[31] F. Chollet, "Xception: deep learning with depthwise separable con-volutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[32] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018, https://arxiv.org/abs/1801.07455.

[33] B.-H. Wang, J. Yu, K. Wang, X.-Y. Bao, and K.-M. Mao, "Fall detection based on dual-channel feature integration," *IEEE Access*, vol. 8, pp. 103443–103453, 2020.

[34] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "An integrated vision-based approach for efficient human fall detection in a home environment," *IEEE Access*, vol. 7, pp. 114966–114974, 2019.

[35] M. Chamle, K. G. Gunale, and K. K. Warhade, "Automated unusual event detection in video surveillance," in *Proceedings of the International Conference on Inventive Computation Technologies. (ICICT)*, pp. 1–4, IEEE, Bangkok, Thailand, August 2016.