Fall Detection Method for Embedded Devices

Xuepei Ma, Xiaohong Wang, and Kun Zhang

Department of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, Hebei 050018, China

E-mail: zhangkun@hebust.edu.cn

Abstract. *The challenges of the aging population is becoming more and more prominent worldwide. Among them, in the face of the elderly fall phenomenon, human fall detection technology research and development has practical application value. Because of a large number of network parameters in the field of fall detection and the limited computing power of embedded devices, which makes it difficult to run on the embedded platform, this paper proposes an OT-YOLOV3 (OpenCV+Tiny-YOLOV3) fall detection method. In this method, Gaussian processing and other operations are used to preprocess the fallen image to avoid the influence of the angle change of the image on the recognition result. Then, the feature extraction network in Tiny-YOLOV3 was replaced by the MobileNet network to increase the number of network layers and reduce the number of parameters and calculations in the model. At the same time, the multi-scale prediction method was used to improve detection accuracy. Experimental results show that the accuracy of the proposed model is 10% higher than that of the YOLOV3 (You Only Look Once Version three) model, 4% higher than that of the Tiny-YOLOV3 model, 3% higher than that of the YOLOV3 model, 3% higher than that of Tiny-YOLOV3 model, and the model size is only 45% of that of YOLOV3 model and 65% of Tiny-YOLOV3. Compared with YOLOV3 and Tiny-YOLOV3 processing methods, the drop recognition effect is significantly improved and the model memory is reduced, which meets the requirements of real-time and efficient detection for embedded devices.* © *2022 Society for Imaging Science and Technology.*

[DOI: 10.2352/J.ImagingSci.Technol.2022.66.4.040407]

1. INTRODUCTION

With continuous development of the economy and the steady improvement in healthcare, the number of births is decreasing year by year and the degree of population aging is deepening. According to the seventh national census of the National Bureau of Statistics, the proportion of the population aged 60 and above in China is 18.70%, of which 13.50% are aged 65 and above. The social pension pressure is also greater. On the face of the lack of pension institutions, home care will be an important trend. Studies indicate that falls are the leading cause of accidental injuries to the elderly [1–3]. Every year, one in three adults aged 65 and older falls down [4]. When there is danger, it is difficult for the elderly to help themselves in time. Without prompt help, falls in the elderly can lead to further disability, serious incapacitation or even death [5]. Therefore, it is very important to detect falling behavior in time [6]. On the one hand, an alarm can be sent to the guardians in time for help, to reduce the injury caused by falling; On the other hand, it can save social public medical resources and reduce the burden of social public medical care. Therefore, the development of automatic fall detection has become an urgent need to protect vulnerable groups, especially the elderly. At the same time, it has also become a research hotspot [3].

Existing fall detection methods are divided into those based on environmental equipment [7], those based on wearable sensors [8], and those based on computer vision technology [9]. Assistant-based falls detection usually relies on some kind of wearable device, which has a variety of sensors. The fall detection method based on computer vision is mainly based on deep learning. In deep learning methods [10], target recognition networks such as YOLO [11] or SSD [12] are used for fall recognition. Although existing deep learning methods has made great achievements in human fall detection, without the 3D Internet, there are limitations in distinguishing falls from similar everyday activities. Moreover, they are heavyweight methods that consume a lot of memory [13], and hence are not suitable for mobile devices. The emergence of Tiny-YOLO [14] solves the problem that embedded devices cannot run to a certain extent. It greatly reduces the layers of the backbone network and reduces the structure and parameters of the model, but the detection accuracy of the network also decreases, making it difficult to achieve a good balance between detection accuracy and real-time performance. Mobile devices are widely used in many scenarios, including the use of mobile devices for fall detection and the development of more economical and lightweight fall detection devices. The above problems limit the application of existing drop detection methods.

Tiny-YOLOV3 is a simplified version of the YOLOV3 network, and its performance comparison between the YOLOV3 model is shown in Table I. In Table I, FLOPS (Floating-point Operations Per Second) represents the number of floating-point operations per second, and Bn represents Billion. Before the network is pruned, compressed, etc, the amount of calculation and weight required by Tiny-YOLOV3 is much smaller than that of YOLOV3, but the model recognition accuracy is lower than that of YOLOV3, mainly because its backbone network is shallow and sacrificed in order to improve speed some accuracy.

To alleviate these problems, combining the advantages of image processing and deep learning network of target

Received Nov. 14, 2021; accepted for publication Mar. 29, 2022; published online May 18, 2022. Associate Editor: Hamad Hamad Naeem.

1062-3701/2022/66(4)/040407/11/$25.00

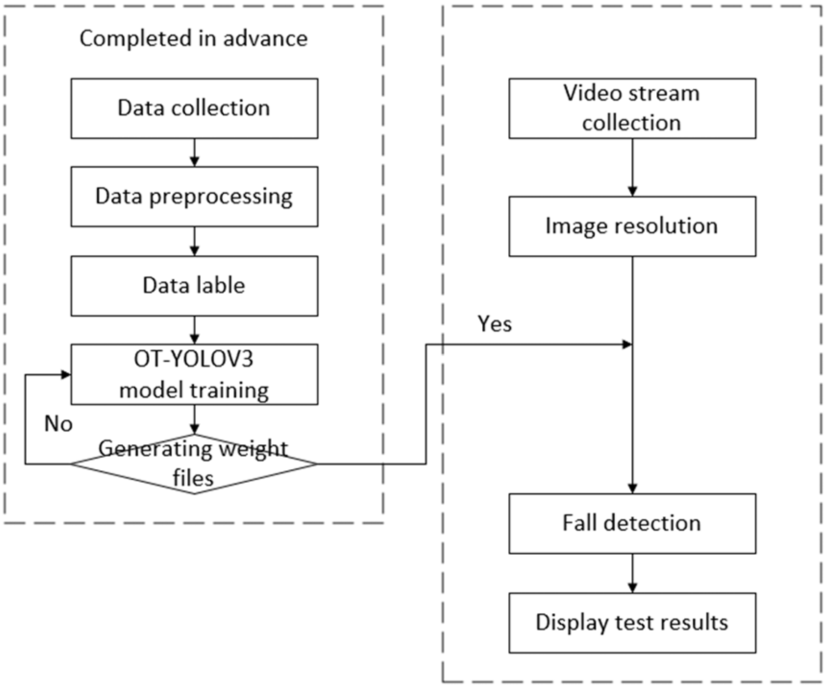

Figure 1. Flow chart of OT-YOLOV3 algorithm.

Table I. Performance comparison between Tiny-YOLOV3 and YOLOV3 models.

A network model	MAP	FLOPS (Bn)	The weight/(MB)
YOLOV3	55.3	65.86	236.52
Tiny-YOLOV3	33.1	5.56	33.79

recognition, this paper proposes a lightweight network of a small model structure and stable to fall detection effect. Firstly, image preprocessing algorithms such as affine transform are used to solve the problem of the declining recognition accuracy caused by the change of shooting angle. Then, the main network of Tiny YOLO was replaced with MobileNet [15], to improve the detection speed to achieve real-time target detection, and the model was transplanted to the embedded platform to meet the computing capacity of the embedded platform. The model structure is modified to increase the number of network layers to improve detection accuracy. The deep separable convolution was used to replace the traditional convolution so that the number of parameters of the model was less, the model was smaller, and the detection accuracy of the model was improved. It is a lightweight model with an efficient fall detection capability. The flow chart of the algorithm is shown in Figure 1.

The data set was collected and annotated in advance, the data set was trained by improving the Tiny-YOLOV3 network model, determines whether a weight file is generated; if so, port to embedded devices; otherwise, retrain the model. Moving the weight file into the embedded device, the device will collect the surrounding information in real-time, the video stream collected and analyzed into images, and then real-time fall detection is finally presented in the results.

Weight files save the weights of all layers of the trained network, that is, the files trained through the constructed fall training set. After the training, the embedded device is implanted, and the application only needs to load the weight, and the training set is no longer needed.

2. RELATED WORK

In the last decade, researchers have done a lot of work on fall detection algorithms. In this chapter, we will review the work related to fall detection from the aspects of wearables, environment deployments, and computer vision.

2.1 *Fall Detection Based on Wearable Sensors*

At present, the sensor used in the fall detection of wearable sensors is usually a triaxial acceleration sensor. By embedding the acceleration sensor into the relevant wearable device, the individual will wear and collect the movement information about the human body. After analyzing and processing the collected information, the decision maker can judge whether there is a fall or not.

Young-Hoon Nho et al. [16] proposed a user-adaptive fall detection method based on cluster analysis that integrates heart rate sensors and accelerometer. An optimal 13 - dimensional feature subset is proposed by the feature selection method. The performance increment of the heart rate sensor combined with the accelerometer and the effectiveness of anomaly detection based on cluster analysis is verified. It demonstrates the effectiveness of a user-adaptive approach using a combination of heart rate and acceleration signals to detect falls. Chen et al. [17] showed that the combination of the integrated stacked autoencoder and the single class classification based on the convex hull is an intelligent fall detection method. The data collected from the accelerometer of the wristband smartwatch is used for unsupervised feature extraction and then for pattern recognition. Using the majority voting strategy and the weight adaptive adjustment strategy, the results obtained from a series of different human behavior characteristics show high performance and stability.

The fall detection device based on the wearable sensor requires the user to wear it before it can be effectively detected. For some elderly people, especially those with severe memory loss, are likely to forget to wear the device, leading to some unpredictable consequences. Some old people may not like to wear this kind of equipment, or feel uncomfortable using it. The battery life of wearable devices is also a consideration.

2.2 *Fall Detection Based on Environment Deployment*

The fall detection technology based on environment deployment generally uses the changes of some physical quantities in the environment caused by human movements, such as vibration, infrared heat source, etc., to identify the movement patterns of the human body by detecting the changes of these physical signals.

Through the combination of PIR and Fresnel lens, Fang et al. identified and detected the target human body walking horizontally and in a straight line in the experimental environment [18–20]. Tan [21] proposed a combination of accelerometers and sound sensors installed at different locations in the home to detect accurate falls and avoid the misclassification of daily activities as falls. A method based on fuzzy logic is used to process the measured signal and interpret the valid drop events. In addition, vibration sensors are also suitable for environmentally deployed fall detection. Alwan's research found that under different movement modes (such as walking, jogging, falling, etc.), the floor would produce vibration of different amplitude and send out signals of different vibration frequencies, and human falls could be detected by monitoring vibration signals [22, 23].

The fall detection system based on environment deployment can only be applied to the indoor environment such as homes, shopping malls, and libraries, and the scope of application is not extensive. The detection system deployed in the environment needs to place the device in the designated position, which makes this method limited by the accuracy and Angle of the auxiliary equipment.

2.3 *Fall Detection Based on Computer Vision*

At present, fall detection based on vision is to use camera devices to collect human video image information, image processing technology is used to process the image information and extract human fall characteristics, and then through the analysis of human motion state.

Ahmad Lotfi et al. [24] proposed a new visual-based fall detection method to support the independent life of the elderly by analyzing the movement and shape of the human body. The motion information was extracted by using the best-fitting ellipse and the enveloping box, and the projected histogram was generated to determine the head position over time. Ten features were generated to identify falls. Xi Cai et al. [25] proposed a visually based fall detection method for multi-task hourglass convolutional autoencoder (HCAE). In this method, the hourglass residuals are introduced into the encoder of HCAE to extract multi-scale features by enlarging the receptive field of neurons. A multi-task mechanism was proposed to enhance the representativity of network features by completing the auxiliary task of frame reconstruction while realizing the main task of fall detection. H. Abdo et al. [26] proposed a fall detection method based on a combination of convolutional neural network RetinaNet and MobileNet and handmade features. The proposed framework relies on RetinaNet to detect the human body, with shorter computation time and higher accuracy. The proposed framework extracts aspect ratio and head position as shape features, and historical motion images as detected human motion features to create feature maps. This feature map was used to train the MobileNet network to classify human movement as falling or non-falling. Chen et al. [27] proposed a method based on attitude estimation and an auxiliary detection method based on yoloV5. Firstly, video frames are extracted from different descending video sequences to form data sets. Then, the training sample set is input into the improved network for training until the network converges. Finally, the classification of the target in the video is tested and the target is located according to the optimized network model.

For the indoor environment with limited space, the surveillance field of view of the camera can be effectively covered, and the fall feature extraction is carried out by an effective image processing method, which has high accuracy and real-time performance. In addition, other additional functions can be expanded based on the visual perception model, such as video monitoring, which can monitor the living conditions of the elderly in real-time through the camera.

To sum up, in practical applications, although deep learning networks can achieve the effect of real-time target detection on large computers such as servers, it is difficult to achieve this effect when the network structure is transplanted to embedded devices such as mobile phones because the processor performance of embedded devices is far inferior to that of servers. Therefore, the network lightweight classification has become an inevitable trend. To alleviate the above problems, this paper proposes a deep learning method of fall detection using OT-YOLOV3.

3. OT-YOLOV3 DEEP LEARNING MODEL

This paper proposes a fall detection and recognition method based on the OT-YOLOV3 deep learning model. Firstly, based on image grayscale and Gaussian blur denoising, the image is binarized and closed open operation, to obtain the image vertex and the image rotation correction, to avoid the impact of image Angle change on the recognition results.

Then, the feature extraction network in Tiny-YOLOV3 is replaced with MobileNet to increase the number of network layers and reduce the number of parameters and computation in the model. Then a target detection model was built based on Keras deep learning framework, and the accuracy and generalization ability of the training model was improved by continuously adjusting parameters and adopting regularization processing. Finally, the categories of falling targets were tested according to the optimized network model.

3.1 *Fall Image Preprocessing*

The image taken by the system is a color image with 256 gray levels. The fallen image may be underexposed or overexposed in the photography, which has the disadvantage of insufficient contrast, and it is unable to identify the important points of the image. The linear expansion of a gray image can optimize the appearance of the image. In this paper, the average gray method is used for processing, it is the three primary color values is added and then divided by the number of primary colors, obtained as the gray value of the point. Its formula is shown in Formula (1).

$$Gray(x, y) = \frac{R(x, y) + G(x, y) + B(x, y)}{3}. \qquad (1)$$

In the formula, x and y are pixel coordinates.

The Gaussian filter is a linear smooth filter, which is suitable for eliminating the regional noise with normal distribution. The mean filter and median filter are widely used in image processing. The Gaussian filter uses the convolution to scan each pixel in the image, and uses the weighted average gray value of the pixels in the neighborhood covered by the convolution kernel to replace the pixel value of its center point. The pixels at different positions in the image neighborhood have different weights, to ensure that the overall features of the image can be retained more while the image noise is processed. The parameters in the convolution kernel are obtained by Eq. (2).

$$G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}. \qquad (2)$$

Among them (x_0, y_0) said the center pixel point, (x, y) said other pixels in the neighborhood.

A binary operation is a process in which the collected image is grayed and the grayscale image is reprocessed to form a process containing only two values, 0 and 1. In the process of image acquisition, due to the impact of illumination factors on the image of the fall, the acquired image has some interference noises, so it is necessary to carry out binarization preprocessing on the image of the fall to reduce the isolated black pixels. In this study, the average gray value of all pixel points in the pixel matrix is calculated to determine the threshold. Its formula is shown in Formula (3).

$$avg = \frac{\sum Gray(x, y)}{n}. \qquad (3)$$

Then compare each pixel with avg one by one, and the pixel less than or equal to avg is 0, and the pixel greater than avg is 255.

The binary image obtained by the threshold method has “noise” and incomplete extraction of the target color region, which will affect the accuracy and reliability of contour detection. Using closed operation and open operation can solve these problems well. Open operation is first corrosion and then expansion, closed operation is the first expansion and then corrosion. The open and closed operations are almost always in the form of “reserved regions”: the most notable effect is that the closed operation eliminates outliers below its neighbors, while the open operation eliminates outliers above its neighbors. In this study, the close operation is firstly used to connect adjacent regions to make the extracted target color region complete, and then the open operation is used to eliminate the noise points in the binary graph.

On the closed operation image obtained after morphological processing, the rotation bounding box of the maximum contour is calculated through findContours function, and the drawContours function makes all connected contour areas in the image add the minimum outer rectangle, and then the image is rotated through affine transformation to avoid the impact of image Angle changes on the recognition results. Its expression is as shown in Formula (4).

$$\boldsymbol{A} = \begin{bmatrix} a_{00}a_{01} \\ a_{10}a_{11} \end{bmatrix}_{2\times 2} \quad \boldsymbol{B} = \begin{bmatrix} b_{00} \\ b_{10} \end{bmatrix}_{2\times 1}. \qquad (4)$$

To use the matrix $\boldsymbol{A}$ and $\boldsymbol{B}$ for 2 d vector $\boldsymbol{X} = \begin{bmatrix} x \\ y \end{bmatrix}$ do change, as shown in formula (5).

$$T = \boldsymbol{A} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \boldsymbol{B}. \qquad (5)$$

The results are shown in Eq. (6).

$$T = \begin{bmatrix} a_{00}x + a_{01}y + b_{00} \\ a_{10}x + a_{11}y + b_{10} \end{bmatrix}. \qquad (6)$$

Affine transformation is mainly used in image translation, scaling, flipping, rotation, cutting and perspective transformation, etc. It can preprocess the image and correct the distorted image, to facilitate identification and segmentation.

3.2 *Fall Detection Model*

For lightweight network problems, compression method based on convolution neural network model, choose MobileNet combined with a Tiny - YOLOV3 detection model, replace the latter with the former network of feature extraction and amend the model structure, increase the network layer to improve detection precision, while retaining the Tiny - YOLOV3 multi-scale prediction, finally be lightweight detection model. The overall technical route structure diagram is shown in Figure 2.

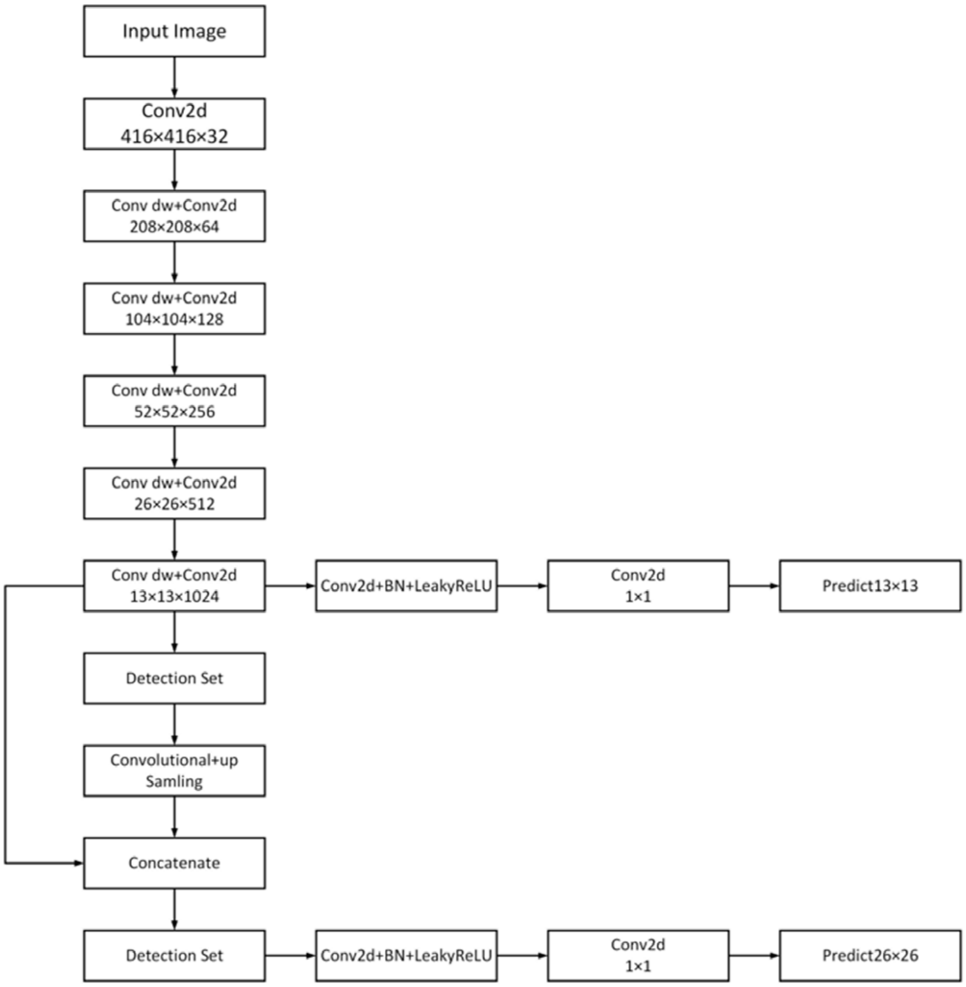

Figure 2. OT-YOLOV3 structure diagram.

The feature extraction network of OT-YOLOV3 adopts 14 3 × 3 convolution kernels for target extraction, of which 5 convolution layers have a step size of 2. The fall image is subsampled to shrink the image. Ten 1 × 1 convolution kernels are used for cross-channel information integration. The number of channels of the feature graph is doubled to make full use of the information of the feature graph every time the image is subsampled with a convolution step of 2.

OT-YOLOV3 uses k-means clustering to generate a priori box, which is mainly set to make the IOU of the prediction box and ground truth better. The dimensions of the 13 × 13 prior frame are (116 × 90), (156 × 198) and (373 × 326), and the dimensions of the 26 × 26 feature map prior frame are (30 × 61), (62 × 45), and (59 × 119). Each grid to predict three bounding boxes, when training model, and at the center of the object in the grid, then only choice and real border IOU overlap degree of the largest bounding box to abandon other smaller values of the IOU bounding box, so that we can improve the model of small target detection ability, raise the bounding box of unit grid generalization ability. Each bounding box outputs the center point coordinates (x, y) of the target frame, height h, width w, and the confidence of the frame. The confidence calculation formula is shown in Formula (7).

$$\text{Confidence} = \Pr(\text{Object}) \times IOU_{\text{pred}}^{\text{truth}}. \tag{7}$$

In the formula: Pr(Object) to predict box contains the probability of the characters, if contain characters, the value is 1, otherwise 0. $IOU_{\text{pred}}^{\text{truth}}$ is the crossover ratio of the prediction box and the real box. Meanwhile, the category confidence of each bounding box prediction is shown in Eq. (8).

$$\Pr(\text{scores}) = \Pr(\text{Class}_i \mid \text{object}) \times \Pr(\text{object}) \times IOU_{\text{pred}}^{\text{truth}}. \tag{8}$$

In the formula: Class_i is species. $\Pr(\text{Class}_i \mid \text{object})$ is the probability that an object belongs to a certain class when there is an object in the prediction box; $i = 1, 2, \ldots, N$, N is the number of categories detected. By setting a threshold, the bounding boxes with a confidence lower than the threshold are filtered out, and the final bounding boxes with confidence higher than the threshold are obtained by non-maximum suppression.

In this paper, the loss function is divided into three parts, namely, the boundary box coordinate error, the boundary box confidence error, and the classification error. The calculation formula of the loss function is.

$$
\begin{aligned}
\text{Loss} = {} & \lambda_c \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& + \lambda_c \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
& + \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} (c_i - \hat{c}_i)^2 \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} (c_i - \hat{x}_i)^2 \\
& + \lambda_{\text{class}} \sum_{i=0}^{S^2} 1_{ij}^{\text{obj}} \sum_{c \in \text{classes}}^{S^2} (p_i(c) - \hat{p}_i(c))^2. \qquad (9)
\end{aligned}
$$

In this formula: λ_c is used to balance three types of losses; $S2$ represents the dimensions of output data; B represents the number of prediction boxes in each cell; 1_{ij}^{obj}represents whether there is a falling figure in the jth prediction box of the ith cell. If there is a falling figure, the value is 1; otherwise, it is 0. Inline first, the central coordinate loss of the prediction box is measured by Euclidean distance. The second row is the width and height loss of the forecast box. Lines third and fourth indicate whether there is a target loss in the prediction box. In general, more boxes do not have targets than boxes that do, and to balance the two, we use $\lambda_{\text{obj}} = 1$ and $\lambda_{\text{obj}} = 0.5$ practice. The last line represents the category loss, λ_{class} controls the balance between class loss and the other two types of losses, and values as 1 in the experiment.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 *Experimental Data and Environment*

In order to use the deep learning method to study the fallen image, it is necessary to establish a relatively comprehensive and systematic fall data set. In this study, a variety of falling gestures are collected through manual shooting, images in public domains and video capturing. About 8000 fall posture pictures were preliminarily collected, from which about 3000 fall and standing pictures with obvious fall characteristics and comprehensive types were selected as the data set of this experiment. Due to the numerous fall modes, the data set is divided into five subsets, side fall, lying fall, prone fall, squatting, and standing. Labelimg tool is used to complete the labeling classification of images, and few data sets of images are shown in Figure 3.

The experimental platform configuration of this project is shown in Table II. The model construction, training, and result testing are all executed under the Keras framework, using the CUDA parallel computing architecture, and the CUDNN acceleration library is integrated into the Keras framework to augment the computing power of the computer.

Table II. Specific environment configuration.

The hardware configuration	System	Ubuntu 16.04
	The graphics card	NVIDIA Ge Force GTX1080
The software configuration	A programming language	Python 3.7
	Deep Learning Framework	Keras
	Annotation tool	LabelImg
	To speed up	CUDA 10.0

4.2 *Experimental Examples of Fall Image Preprocessing*

To highlight the contour features of the image, the average grayscale method is used for grayscale processing. The conversion effect of the average grayscale method is better than other methods. The original input results are shown in Figure 4, and the grayscale results are shown in Figure 5. It is found in the experimental process that it is easier to extract the contour by using a Gaussian filter to smooth the gray image. In the experiment, a 9 × 9 convolution kernel is adopted, and then the image is binarized. The processing results are shown in Figures 6 and 7.

Because of the problem of “noise” in the binarization image, closed operation and open operation are adopted for the binarization image, to make the extracted target color region complete and eliminate the noise in the binarization image. The results of the closed-open operation are shown in Figures 8 and 9.

After the image is processed by open operation, the image edge information is extracted, the external contour is calculated, the minimum external rectangle is added, and then the affine transformation is carried out to solve the problem of shooting angle. The result of contour extraction is shown in Figure 10, and the result of the affine transformation is shown in Figure 11.

After image grayscale; Gaussian filtering, binarization, closed-open operation, contour extraction, and affine transformation operation can be used for distorted fall image correction, easy identification, segmentation and other operations.

4.3 *Experimental Examples of Fall Detection*

Using the Tiny-YOLOV3 target detection network, the regularization method was adopted to increase the generalization ability of the model during the training process and ensure the balance of the samples of each fall mode. Parameters were fine-tuned according to the self-made training set to achieve the best training effect. Some test parameters were adjusted as shown in Table III. The fall state recognition results are shown in Figures 12–16.

4.4 *Evaluation Indicators*

Given the prediction effect of the model, three common indicators of deep learning – accuracy, recall rate, and average precision (AP) were used to measure the results of the experiment. The fall detection includes five cases of standing, squatting, lying down, falling on the side, and

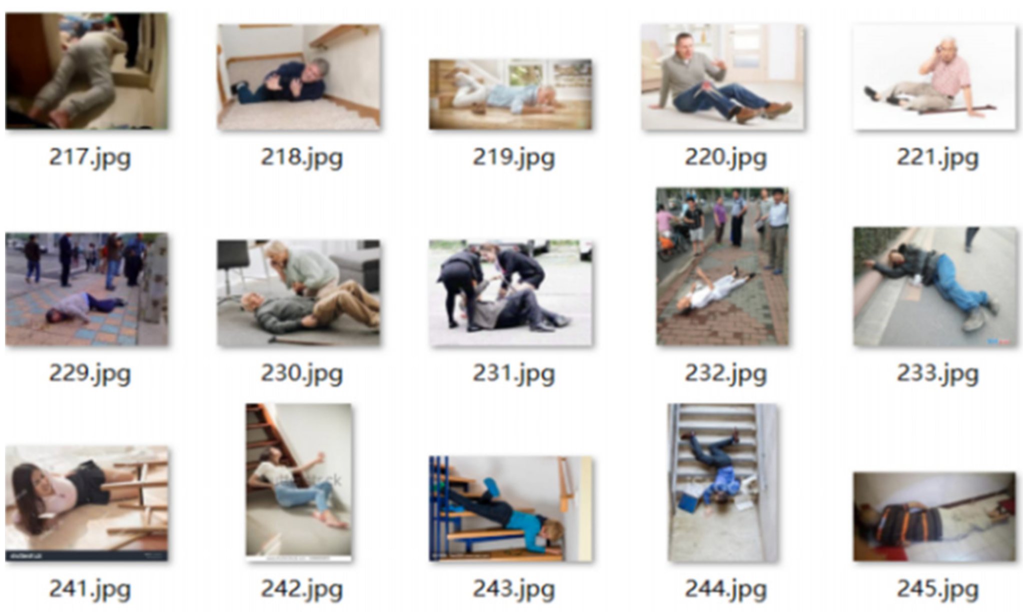

Figure 3. Partial dataset in a real world scenario.

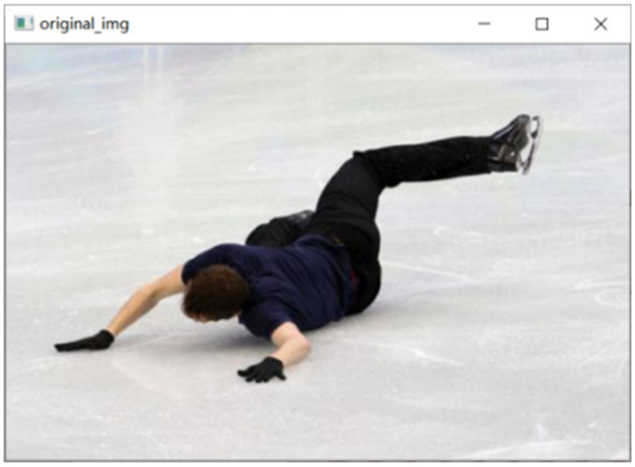

Figure 4. Raw image.

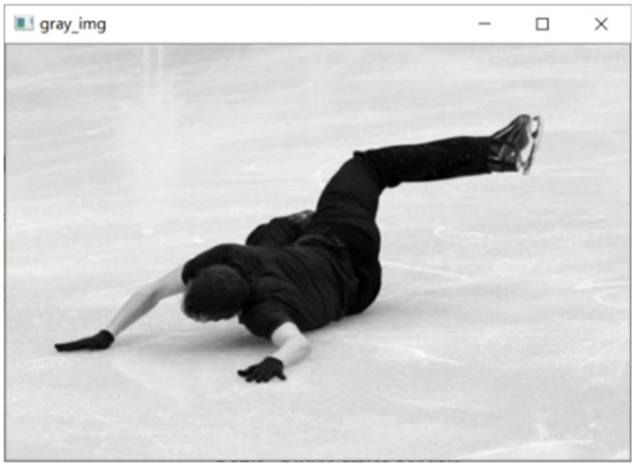

Figure 5. Grayscale result.

Table III. Network parameter description table.

Parameter names	Parameter value
Learning rate	0.001
Momentum optimizer	0.9
Decay factor	0.95
Batch size	64

Table IV. Meaning of TP/TN/FP/FN index.

		Actually	
		1	0
Predict	1	TP True Positive	FP False Positive
	0	FN True Negative	TN True Negative

falling. Therefore, the mean average precision (mAP) of the five categories is adopted as the final evaluation index of the model, and the calculation method is as follows. As shown in Table IV, TP represents the feature number of predicted positive samples and actually positive samples; TN represents the feature number of predicted negative samples and actually negative samples; FP represents the feature number of predicted positive samples and actually negative samples; FN represents the feature number of predicted negative samples and actually positive samples.

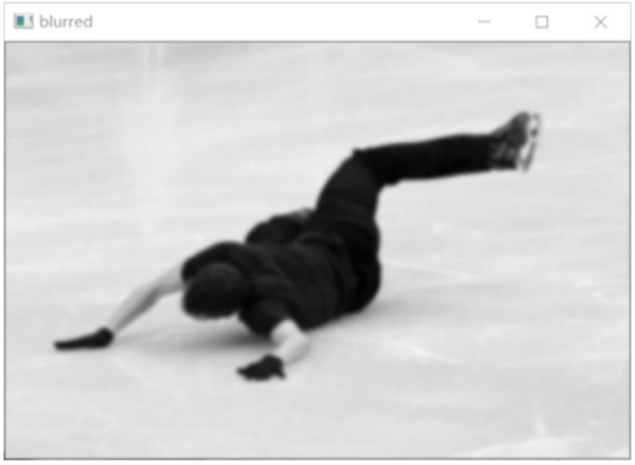

Figure 6. Gaussian filtering results.

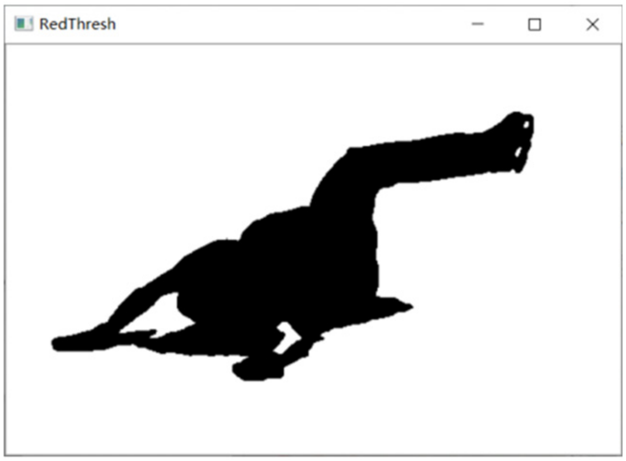

Figure 7. Binarization results.

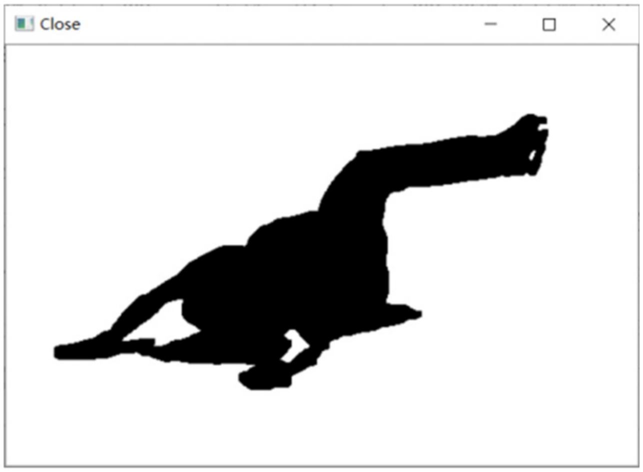

Figure 8. Closed operation results.

Accuracy rate and:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{10}$$

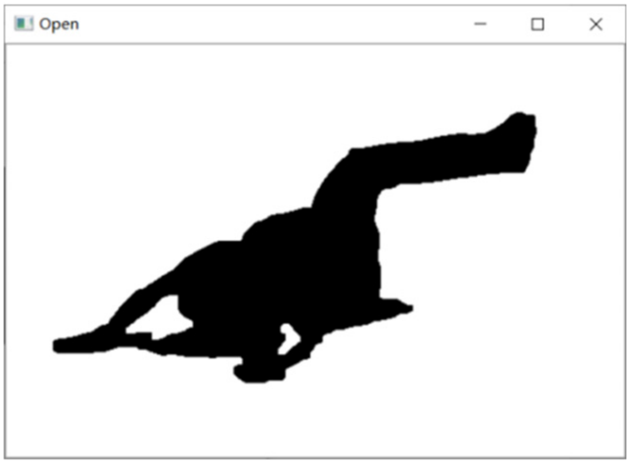

Figure 9. Open-operation results.

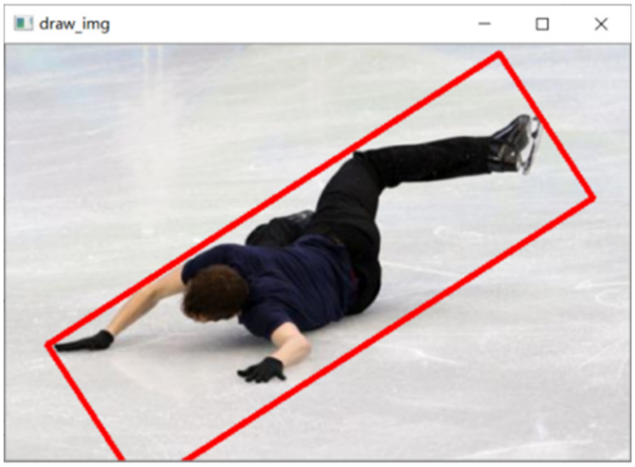

Figure 10. Result of contour extraction.

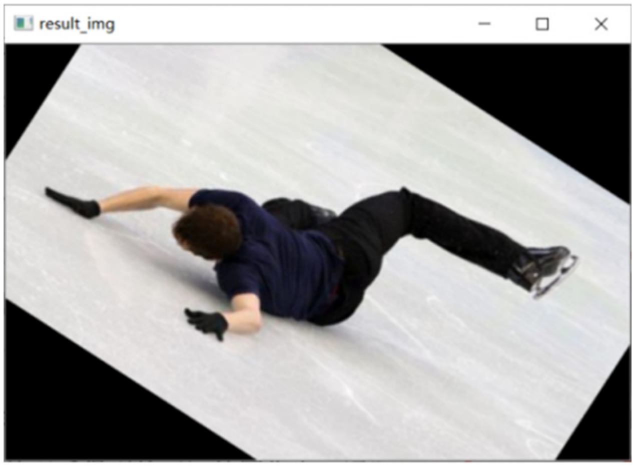

Figure 11. Affine transformation.

The recall rate:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{11}$$

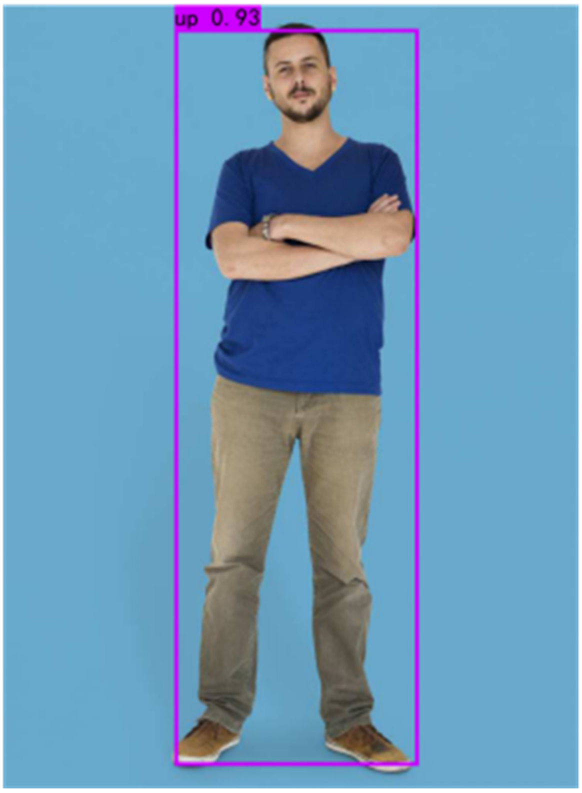

Figure 12. Standing results.

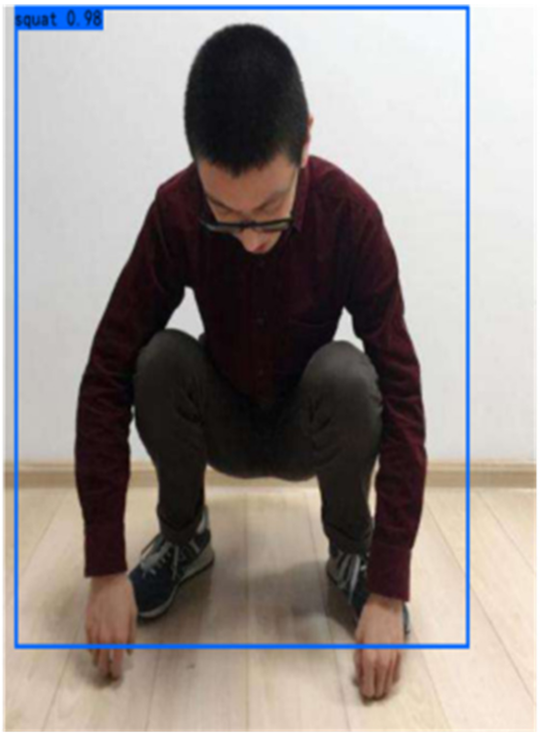

Figure 13. Squat results.

The average precision:

$$\mathrm{AP} = \int_0^{\mathrm{recall}} \mathrm{precision} \tag{12}$$

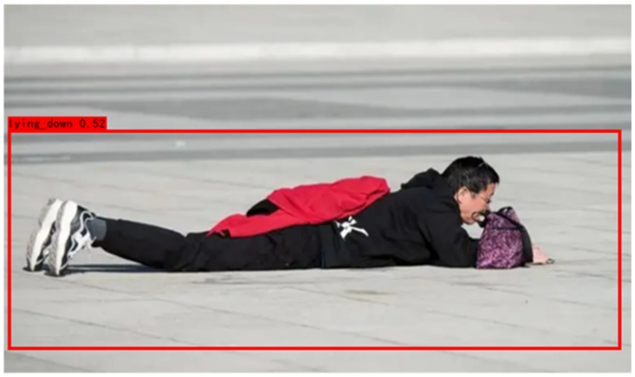

Figure 14. Falling results.

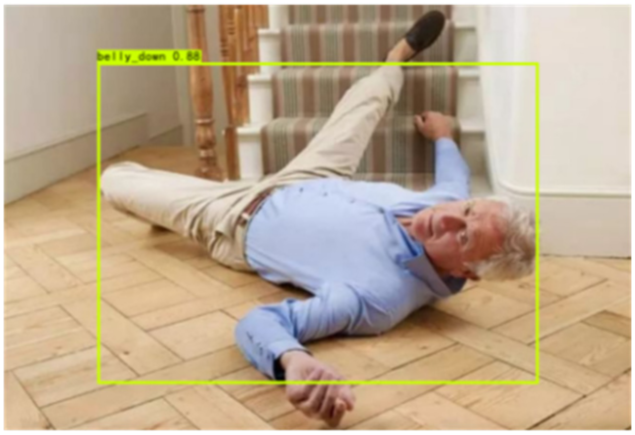

Figure 15. Results of lying down and falling.

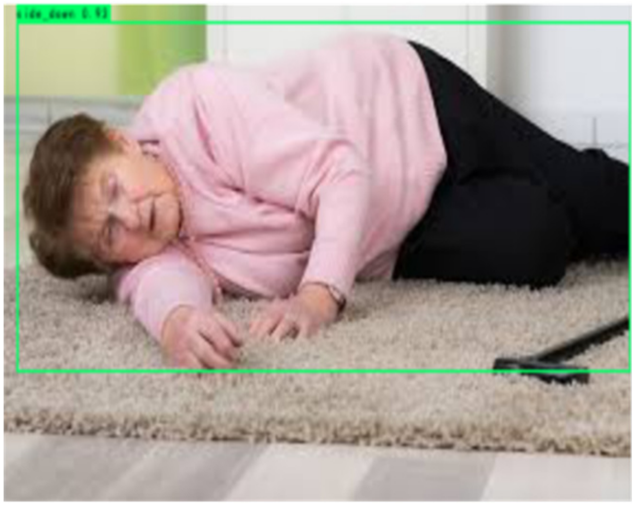

Figure 16. Side fall results.

The mean average precision:

$$\mathrm{mAP} = \frac{1}{m}\sum_{i=0}^{m} \mathrm{AP}_i. \tag{13}$$

Table V. Performance comparison of learning models.

Model	YOLOV3	Tiny-YOLOV3	OT,-YOLOV3
Accurate rate (Map)	80%	86%	90%
The recall rate	81%	84%	87%
Detection speed	118.54	101.38	94.14
The model size	49	34	22

As can be seen from Eqs. (10) and (11), the accuracy rate is based on the prediction result, indicating how many of the predicted positive samples are actually positive samples; The recall rate is for the actual sample, indicating how many of the actual positive samples were also predicted to be positive. It can be seen from Eqs. (12) and (13) that AP is the average accuracy of each target type and M is the total number of target types.

In the test set, the model used in this study, the Tiny-YOLOV3 method and the classification method using YOLOV3 were tested respectively, and the recognition accuracy, recall rate, and recognition speed were determined and is shown in Table IV.

As can be seen from Table V, the detection results of the OT-YOLOV3 deep learning model are significantly higher than those of YOLOV3 and Tiny-YOLOV3, and the detection effect is 10 percentage points higher than that of YOLOV3 MAP and 4 percentage points higher than that of Tiny-YOLOV3 MAP. Meanwhile, the model size of the proposed algorithm in the self-built data set is 22MB, which is 45% of the size of the YOLOV3 model and 65% of the size of the Tiny-YOLOV3 model. The model memory is reduced while the accuracy is improved. However, the complexity of the deep learning model increases its running time, and this method is slightly slower than the other two methods in terms of computation time. But in actual industrial production, the speed difference is still within the permissible range. Formula 13 can be used to obtain the average detection accuracy of five targets by different models. See Figure 17. As can be seen from the figure, the accuracy of the method proposed in this paper is higher than that of other models in each category recognition. The performance of the model has improved remarkably.

Table VI is a comparison of the detection speed of different models. It can be seen from Table VI that the frame rate of OT-YOLOV3 is 243.6f/s, which is higher than that of YOLOV3 and slightly lower than that of Tiny-YOLOV3. This is because the OT-YOLOV3 model deepens the number of network layers, which reduces the detection speed, while the introduction of MobileNet can effectively enhance the expression ability of complex models and reduce the amount of information processed. The comparison results show that OT-YOLOV3 not only ensures the improvement of detection accuracy, but also has less impact on the detection speed, and can obtain better detection outcomes.

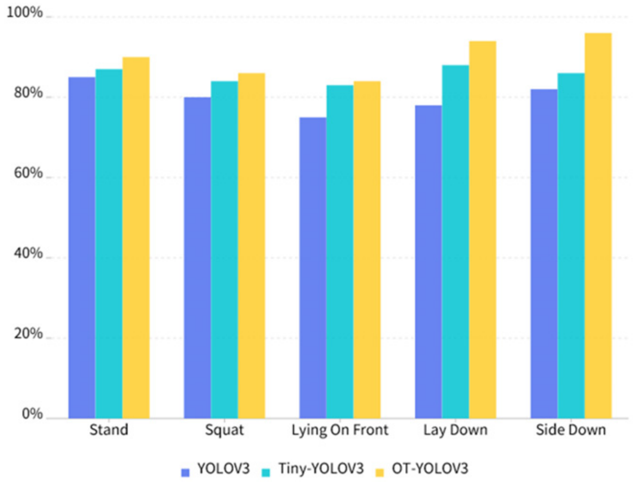

Figure 17. Average detection accuracy of different models for five targets.

Table VI. Comparison of detection speed of different models.

Model	YOLOV3	Tiny-YOLOV3	OT-YOLOV3
Total detection time/ms	8563.7	3456.6	4123.8
Mean detection time/ms	25.2	4.4	6.0
Frame rate $(\mathrm{f} \cdot \mathrm{s}^{-1})$	74.3	340.5	243.6

The application scenarios of this method are very wide. For instance, in the field of intelligent transportation, the model of an intelligent traffic light control system can be constructed. Firstly, the OpenCV library function is called to divide the video acquisition area and identify and count the areas. Then, the model is deployed to STM32 to detect and determine the relative position and number of vehicles in the image for the recognition algorithm and positioning algorithm, and the passing time of changing the left turn and going straight is given, to achieve intelligent control. The method described in this paper can also be applied to tumor cell detection. OpenCV is used to de-noise the nuclear image, followed by training using the model network, and the results are obtained.

5. CONCLUSION

OT-YOLOV3 deep learning model for fall detection was proposed for timely detection of falls in old people and to solve the problems that the embedded device model with weak computing power and low detection accuracy in the field of fall detection. The fall image is preprocessed by grayscale, Gaussian filtering, binarization, closed-open operation, contour extraction, and other pre-processing operations. In this model, the feature extraction network of Tiny-YOLOV3 is replaced by MobileNet, which reduces the number of parameters and model size. The detection accuracy of the model was improved by generating the prior frame by K-means clustering; feature splicing and multi-scale prediction methods were adopted to ensure the fall detection effect of the model for multiple types of different poses. The experimental results show that the accuracy of the proposed algorithm is up to 90%, which has a definite advantage over YOLOV3 and Tiny-YOLOV3, through improved accuracy and reduced calculation amount of the model. The proposed algorithm is robust, portable in the actual environment, and can accurately identify falls. Future work can improve the detection speed of the model on the premise of ensuring accuracy.

ACKNOWLEDGMENT

This work was supported in part by the Department of Education of Hebei Province under Grant ZD2020176, Hebei Science and Technology Department project 21470302D.

REFERENCES

[1] C. Taramasco, T. Rodenas, F. Martinez, P. Fuentes, R. Munoz, R. Olivares, V. H. C. De Albuquerque, and J. Demongeot, "A novel monitoring system for fall detection in older people," IEEE Access **6**, 43563–43574 (2018).

[2] D. France, J. Slayton, S. Moore, H. Domenico, J. Matthews, R. L. Steaban, and N. Choma, "A multicomponent fall prevention strategy reduces falls at an academic medical center," Jt. Comm. J. Qual. Patient Saf. **43**, 460–470 (2017).

[3] P. Youngkong and W. Panpanyatep, "A novel double pressure sensors-based monitoring and alarming system for fall detection," *2021 Second Int'l. Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)* (IEEE, Piscataway, NJ, 2021), pp. 1–5.

[4] G.-M. Sung, H.-K. Wang, and W.-T. Su, "Smart home care system with fall detection based on the android platform," *2020 IEEE Int'l. Conf. on Systems, Man, and Cybernetics (SMC)* (IEEE, Piscataway, NJ, 2020), pp. 3886–3890.

[5] Y. N. Chen, C. H. Chuang, H. M. Lee, C. C. Yu, and K. C. Fan, "Fall detection in dusky environment," J. Image Video Proc. **16** (2016).

[6] I. N. Figueiredo, C. Leal, L. Pinto, J. Bolito, and A. Lemos, "Exploring smartphone sensors for fall detection," mUX J. Mob. User Exp. **5**, 2 (2016).

[7] S. Rashmi and P. Manju, "Adaptive window based fall detection using anomaly identification in fog computing scenario," Multiagent Grid Syst. **17** (2021).

[8] F.-S. Hsu, T.-C. Chang, Z.-J. Su, S.-J. Huang, and C.-C. Chen, "Smart fall detection framework using hybridized video and ultrasonic sensors," Micromachines **12** (2021).

[9] S. Liu, Z. An, N. Wang, D. Bai, and X. Yu, "Research on elevator passenger fall detection based on machine vision," IOP Conf. Ser.: Earth Environmental Sci. **791** (2021).

[10] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamara, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," J. Big Data **8**, 53 (2021).

[11] X. Liao and X. Zeng, "Review of target detection algorithm based on deep learning," *Proc. 2020 Int'l. Conf. on Artificial Intelligence and Communication Technology (AICT 2020). Institute of Management Science and Industrial Engineering: Computer Science and Electronic Technology International Society* (Clausius Scientific Press, Canada, 2020), pp. 62–66.

[12] T. Li, H. Wu, Y. Mao, D. Li, and L. Chen, "Survey and application of target detection algorithms based on deep learning," World Sci. Res. J. **6** (2020).

[13] R. Kamiya, T. Yamashita, M. Ambai, I. Sato, Y. Yamauchi, and H. Fujiyoshi, "Binary-decomposed dcnn for accelerating computation and compressing model without retraining," *Proc. IEEE Int'l. Conf. Comput. Vis. Workshops (ICCVW), Venice, Italy, Oct.* (IEEE, Piscataway, NJ, 2017), pp. 1095–1102.

[14] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," J. Big Data **8**, 66 (2021).

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyang, M. Andreetto, and H. Adam, "MobileNets: efficient convolutional neural networks for mobile vision applications [EB]" Preprint arXiv:1704.04861 (2017).

[16] Y.-H. Nho, J. G. Lim, and D.-S. Kwon, "Cluster-analysis-based user-adaptive fall detection using fusion of heart rate sensor and accelerometer in a wearable device," IEEE Access **8**, 40389–40401 (2020).

[17] L. Chen, R. Li, H. Zhang, L. Tian, and N. Chen, "Intelligent fall detection method based on accelerometer data from a wrist-worn smart watch," Measurement **140**, 215–226 (2019).

[18] J. S. Fang, Q. Hao, D. J. Brady, B. D. Guenther, and K. Y. Hsu, "A pyro-electric infrared biometric system for real-time walker recognition by use of a maximum likelihood principal components estimation (MLPCE) method," Opt. Express **15**, 3271–3284 (2007).

[19] J. S. Fang, Q. Hao, D. J. Brady, M. Shankar, B. D. Guenther, N. P. Pitsianis, and K. Y. Hsu, "Path-dependent human identification using a pyroelectric infrared sensor and Fresnel lens arrays," Opt. Express **14**, 609–624 (2006).

[20] J. S. Fang, Q. Hao, D. J. Brady, B. D. Guenther, and K. Y. Hsu, "Real-time human identification using a pyroelectric infrared detector array and hidden Marko models," Opt. Express **14**, 6643–6658 (2006).

[21] P. V. Er and K. K. Tan, "Non-intrusive fall detection monitoring for the elderly based on fuzzy logic," Measurement **124**, 91–102 (2018).

[22] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," *In 2nd Int'l. Conf. on Information & Communication Technologies* (IEEE, Piscataway, NJ, 2006), 10031007.

[23] 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," Biomed. Eng. **56**, 28582867 (2009).

[24] A. Lotfi, S. Albawendi, 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 **6**, 70272–70282 (2018).

[25] X. Cai, S. Li, X. Liu, and G. Han, "Vision-based fall detection with multi-task hourglass convolutional auto-encoder," IEEE Access **8**, 44493–44502 (2020).

[26] H. Abdo, K. M. Amin, and A. M. Hamad, "Fall detection based on RetinaNet and MobileNet convolutional neural networks," *2020 15th Int'l. Conf. on Computer Engineering and Systems (ICCES)* (IEEE, Piscataway, NJ, 2020), pp. 1–7.

[27] Y. Chen, R. Du, K. Luo, and Y. Xiao, "Fall detection system based on real-time pose estimation and SVM," *2021 IEEE 2nd Int'l. Conf. on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)* (IEEE, Piscataway, NJ, 2021), pp. 990–993.