Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4110246, 10 pages
https://doi.org/10.1155/2022/4110246

Research Article

Pedestrian Fall Event Detection in Complex Scenes Based on Attention-Guided Neural Network

Peng Geng, Hui Xie, Houqin Shi, Rui Chen, and Ying Tong

School of Information and Communication Engineering, Nanjing Institute of Technology, Nanjing, China

Correspondence should be addressed to Rui Chen; j00000002555@njit.edu.cn

Received 8 February 2022; Revised 11 March 2022; Accepted 1 April 2022; Published 28 April 2022

Academic Editor: Wei Liu

To address automatic detection of pedestrian fall events and provide feedback in emergency situations, this paper proposes an attention-guided real-time and robust method for pedestrian detection in complex scenes. First, the YOLOv3 network is used to effectively detect pedestrians in the videos. Then, an improved DeepSort algorithm is used to track by detection. After tracking, the authors extract effective features from the tracked bounding box, use the output of the last convolutional layer, and introduce the attention weight factor into the tracking module for final fall event prediction. Finally, the authors use the sliding window for storing feature maps and SVM classifier to redetect fall events. The experimental results on the CityPersons dataset, Montreal fall dataset, and self-built dataset indicate that this approach has good performance in complex scenes. The pedestrian detection rate is 87.05%, the accuracy of fall event detection reaches 98.55%, and the delay is within 120 ms.

1. Introduction

Pedestrian fall event detection is one of the challenging problems for public security, particularly in some crowded complex environments. Fall events are also the leading factor of physical injury among elderly. In the WHO report of 2020, fall-related mortality rate is 6%, so there has been much research interest in fall-alerting systems. Many research studies based on a variety of devices, such as wearable devices [1, 2] and imaging sensors [3], have been presented to detect fall events. The works based on wearable devices, including tilt sensors, accelerometers, and gyroscopes, achieve good detection performances. But it is impossible for people to wear specific equipment in crowded situations. Imaging sensor-based approaches depend on cameras, depth sensors, and infrared sensors. These methods and datasets mainly focus on detection in indoor scenarios equipped with expensive devices, and complex scenarios are not considered.

At present, various vision-based pedestrian fall detection methods have been developed and many existing problems have been solved. These proposed methods can be classified into two categories, namely, two-stage method and one-stage method. The two-stage method is based on regional proposal, and its typical methods include Girshick’s R-CNN [4] and R-CNN’s various improved versions [5, 6]. As the extraction of region proposal is time consuming, even in faster R-CNN, the alternative training is still required to get shared convolutional parameters between the region proposal network and the detection network. Therefore, processing time becomes a bottleneck for real-time applications. The one-stage method is based on regression, such as YOLO (You Look Only Once) [7], SSD (Single Shot MultiBox Detector) [8], and their variants. This method has fast detection speed, but it is difficult in small target grouping processing. The above generic object detection approaches achieve the most advanced performance on the benchmark dataset. Due to lots of small-scale pedestrian instances existing in typical scenes of pedestrian detection, the application of ROI (region of interest) pool layer in the general target detection pipeline will lead to “plain” feature caused by the collapse of the dustbin. Many researchers have conducted studies to adapt generic detector to detect pedestrians. On the basis of faster R-CNN, Zhang et al. [9] revised the downstream classifier via introducing enhanced forest into the shared high-resolution convolution feature map and using region proposal network (RPN) to process small-scale objects and hard negative samples. For occlusion problem, Dinakaran et al. [10] presented a deep learning framework to deal with partial complex occlusions, and judgments are made according to several partial detectors. As LSTM (long short-term memory) can derive temporal information from video sequence by exploiting the fact that feature vectors are connected semantically for contiguous frames, various CNN-LSTM models are proposed to obtain spatial-temporal information for better detection performance. A cascaded LSTM [11] is presented for training several partial detectors to handle the common occlusion patterns and integrated into the detection module. Since the attention mechanism can quickly focus on regions of interest in complex scenes, some approaches introduced attention mechanism into the fall event detection framework. Qi et al. [12] proposed an explicit attention-guided LSTM based framework of pedestrian fall event detection, in which YOLOv3 is used to detect pedestrians in video frames, DeepSort algorithm [13] is used to complete the tracking task, and VGG-16 is used to extract the features from the tracked bounding boxes. For occluded pedestrian detection, Zou et al. [14] proposed an attention-guided deep learning network to handle the occluded problem, which integrated the CNN, attention mechanism module, and RNN into one framework. The attention module is used to guide LSTM to generate the feature representation, so the performance deterioration caused by occlusions can be greatly decreased. Although these methods can effectively detect pedestrian fall events, the training process and RPN are very time consuming. Zhou and Yuan [15] presented a joint learning algorithm to train the part detectors and reduce training time. But the detection rate relies heavily on the occlusion pattern.

For pedestrian fall event detection, the target has the characteristics of large posture changes and fast speed. Utilizing these characteristics, a fall event detection method [16] is proposed, which is based on the finite state machine theory. However, the detection performance of this method is highly dependent on the aspect ratio, which leads to weak robustness. According to angle and distance information, Chua et al. [17] classified the fall events by the changes of posture state. Many fall event detection algorithms based on neural network have been proposed, such as PCANet [18], two-stream CNN-based action detection [19], and so on. These methods have good performance in fall event detection in solitary scene, but in complex scenes, that is, when there are severe occlusion, insufficient lighting, and scale changes, they are difficult to locate the fall event.

Focusing on fall event detection in complex environments, the authors present an attention-guided fall event detection algorithm to handle occlusion, illumination change, and scale change. The authors add an attention-guided neural network to the YOLOv3 network, which can effectively solve the problem of losing targets due to occlusion. The rest of this paper is organized as follows. Section 2 introduces the framework and implementation details of the proposed fall event detection method. The experimental results are described and analyzed in Section 3. The conclusion of this paper is given in Section 4.

2. The Proposed Algorithm

A new attention-guided algorithm is proposed in this paper, which is used to detect fall events in complex scenes, and its framework is depicted in Figure 1. It includes three modules: pedestrian detection, target tracking, and redetection modules. The pedestrian detection module includes two branches, one is the traditional YOLOv3 network, and the other is block-based feature extraction and attention module. The attention module guides the neural network to generate an attention weight factor. The target tracking module is used to track each pedestrian for the trajectory which contains continuous event in the video sequence. The tracking module uses the DeepSort [13] algorithm to track by detection. The redetection module uses the sliding window for storing feature maps and SVM classifier to redetect fall events.

2.1. Pedestrian Detection. Tracking-by-detection is a multi-target tracking method, and selecting an appropriate and excellent detector has a great impact on the tracking effect. YOLO [20] is a target detection algorithm based on one stage, which processes and learns the target region, position, and class of the corresponding target at one time by means of direct regression. Many YOLO-based approaches have been proposed for pedestrian detection [21–23]. YOLOv3 [24] can predict 4 coordinated values for each bounding box (t_x, t_y, t_w, t_h). Let P_W be the width and P_H be the height of bounding box; based on the deviation of the upper left corner of image (c_x, c_y), next bounding box can be predicted by

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = P_W e^{t_w} \\ b_h = P_H e^{t_h} \end{cases}. \tag{1}$$

The structure of YOLOv3 is depicted in Figure 2. It includes a feature extraction module and a detection module. The former integrates YOLOv2, Darknet-53, and ResNet. Unlike the traditional CNN [5], Darknet-53 discards the commonly used pooling layers and carries the Leaky-ReLU activation function after convolutional layer. Also, no bias is utilized in Leaky-ReLU function's input, which can simplify the model and reduce the dimension and parameters of the convolution kernel. Furthermore, the feature extraction capability of the model is enhanced, and the timeliness and sensitivity of pedestrian detection are improved.

To detect small crowded targets with low resolution, the authors use multi-scale prediction. The prediction is implemented on three scales with the strides of 52, 26, and 13, respectively. Based on multi-scale characteristics of the network, the convolutional layer of different receptive fields in the network is improved to be used as a separate output

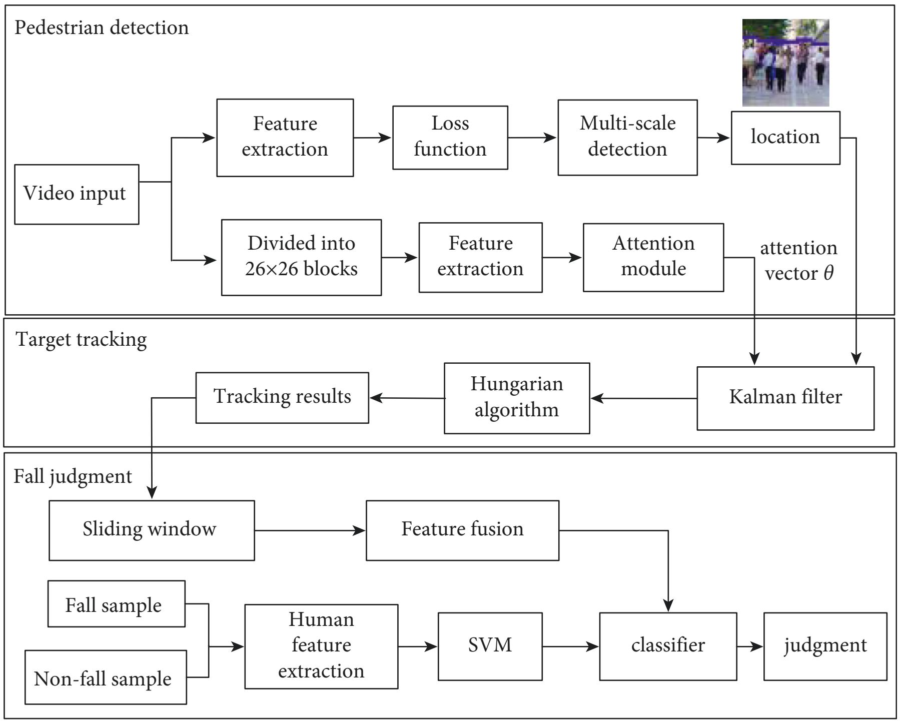

FIGURE 1: Framework of the proposed attention-guided fall event detection method.

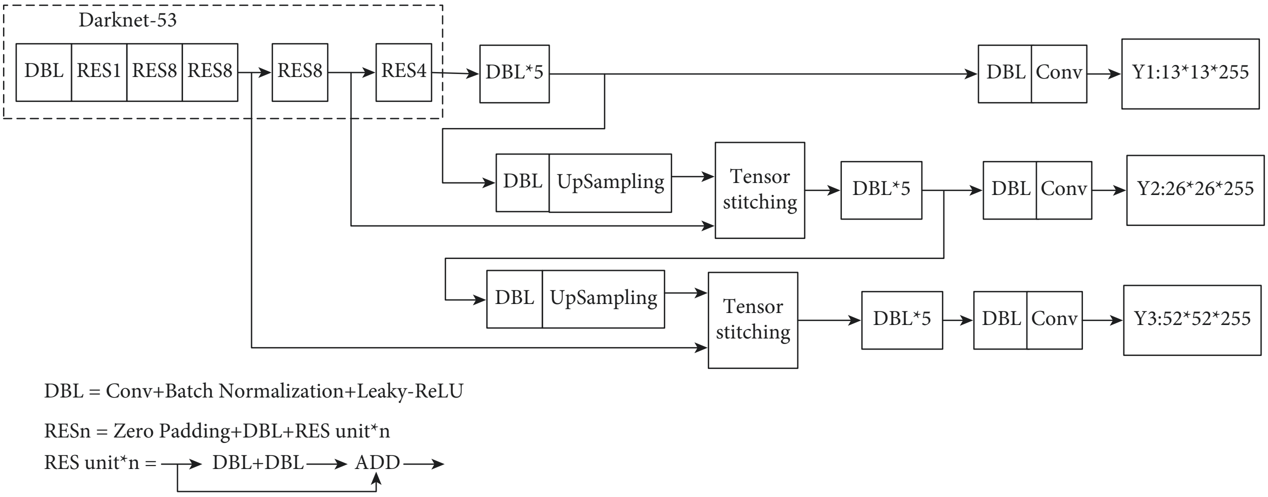

FIGURE 2: YOLOv3 network structure diagram.

for classification calculation. In addition, the network can adjust the priori box of the receiving field convolutional layer according to the value of GT (ground truth). IoU (intersection over union) is calculated by

$$\mathrm{IoU} = \frac{\mathrm{area}\,(A) \cap \mathrm{area}\,(B)}{\mathrm{area}\,(A) \cup \mathrm{area}\,(B)}, \tag{2}$$

where area (A) represents the GT bounding box area and area (B) represents the candidate bounding box area. IoU close to 1 means that the candidate and the ground truth bounding boxes overlap completely. Finally, because different distances between pedestrians and cameras will produce size differences, the detection box can be resized accordingly.

The distance between the prediction and the real bounding box is estimated by using the loss function in YOLOv3, which is multi-objective, including localization error, confidence error, and classification error:

$$\mathrm{Loss} = l_{\mathrm{xy}} + l_{\mathrm{wh}} + l_{\mathrm{cls}} + l_{\mathrm{conf}}, \tag{3}$$

where l_{xy} and l_{wh} are localization errors calculated by sum of error square loss function and l_{conf} and l_{cls} are confidence error and classification error, respectively, calculated by binary cross entropy loss function.

2.2. Attention-Guided Tracking Module. In order to handle occlusions in pedestrian detection, the authors introduce the spatial attention module to increase the feature weight of pedestrian's body parts (such as head, trunk, feet, and so on), so that the tracker can focus on these key body parts which can avoid the influence of interference information such as background occlusion. The attention module needs to use a fixed window of 26×26 first and divide the static 416×416 image into 16 subimages by sliding from left to right for odd lines and from right to left for even lines. Then, CNN is used to extract features from these subimages to obtain a series of feature sequences.

The attention module is shown in Figure 3. First, each video frame is segmented into N subimages, denoted as $s(t)\,(t = 1, 2, \ldots, N)$. The features of subimage sequence are extracted by CNN, denoted as $f(t)\,(t = 1, 2, \ldots, N)$, and introduced to the attention module for generating the attention vector θ.

Attention module implements the local feature weighting and learns a mapping function F for regressing the attention vector θ as

$$\theta_t = \frac{\exp(F(x(t)))}{\sum_{i=1}^{N} \exp(F(x(i)))}, \quad t = 1, 2, \ldots, N. \tag{4}$$

The size of θ_t represents the probability of whether $f(i)$ is a body part feature, and the element weighting of F is as follows:

$$F(i) = f(i) \odot \theta_i, \quad i = 1, 2, \ldots, N, \tag{5}$$

where $F(i)$ represent weighted elements of F and $\odot$ represents the element-wise multiplication. Through continuous learning of the attention module, the authors can update its parameters and optimize the weight θ_t in (4). The attention vector θ_t is introduced into tracking module to improve the detection and tracking efficiency of pedestrians.

For pedestrian tracking, the DeepSort method [13] is used to predict the next position of each trajectory. First, the Kalman filter is used to obtain the features of the extracted targets in the previous frame, including the center position coordinates, the aspect ratio, the height, and the speed. Then, the next location $\hat{X}_K$ is predicted by using the error covariance matrix, and it can be corrected by $\hat{X}_K = K_K \times Z_K + (1 - K_K) \times \hat{X}_{K-1}$, where K_K is the Kalman gain and Z_K is the actual measured value which can be corrected by $\hat{X}_{K-1}$ and K_K. The optimal estimation is the prediction.

The DeepSort algorithm is based on sort algorithm. It has the characteristics of deep correlation and conducts tracking task by the exact detection results. The DeepSort algorithm takes the detection results, bounding box, confidence, and feature as inputs, where confidence is mainly used for filtering detection boxes and bounding box and feature (ReID) are used for matching calculation with tracker. The prediction task is completed by the Kalman filter, and the update part adopts IoU to match the Hungarian algorithm. A tracking scenario is defined by an eight-dimensional state space $(\mu, \nu, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$, where (μ, ν) represents the center of the bounding box, γ represents the rectangular aspect ratio of the target, h represents the height of the bounding box, and $(\dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ describes the motion feature. The algorithm applies the standard Kalman filter of the linear observation model and uniform model to calculate the target trajectory in the following frame and takes the boundary coordinates (μ, ν, γ, h) as the direct observation of the object state. For each trajectory, the authors record the number of frames between the last successfully detected frame and the current detected frame as a_k. The counter is incremented during Kalman filter prediction and set to zero when the trajectory is associated with the measurement. The value of a_k exceeding the threshold $A_{\max}$ means that the trajectory has lost and the target is out of the scene, so the trajectory is removed. If no detection can match the existing trajectory in the detector, then the detector will generate a tentative trajectory. If a trajectory cannot be rematched in 3 frames, then it will be removed.

The sort tracking algorithm was first proposed in [25], aiming at real-time online tracking. When the target is occluded or missed in multiple frames, the trajectory of the same target will be suspended and a new one will be generated. The DeepSort method solves this problem, and it combines the Mahalanobis distance and cosine distance metrics to obtain the final decision information $C_{i,j}$ by weighted summation:

$$C_{i,j} = \lambda d_1(i, j) + (1 - \lambda) d_2(i, j), \tag{6}$$

where λ is a superparameter which is used to adjust the weight of different items, $d_1(i, j)$ represents the Mahalanobis distance, and $d_2(i, j)$ represents the cosine distance. The DeepSort algorithm uses the Kalman filter to calculate next position for every trajectory and then calculates the Mahalanobis distance $d_1(i, j)$ by

$$d_1(i, j) = \left(d_j - y_i\right)^T S_i^{-1} \left(d_j - y_i\right), \tag{7}$$

where d_j represents the location of j-th bounding box, y_i represents the prediction of the target location from i-th tracker, S_i represents the covariance matrix between the location of detection and tracking, and (y_i, S_i) describes the projection of the i-th tracker to the measurement space. Considering the state estimation uncertainty, the Mahalanobis distance measurement detects the standard deviation from the average track position. It retains the result of spatial distribution and is more efficient, while in order to express the correlation degree of appearance features, the least cosine distance between i-th and j-th can be calculated as

$$d_2(i, j) = \min\left\{1 - r_j^T r_k^{(i)} \middle| r_k^{(i)} \in R_i\right\}, \tag{8}$$

where r_j is the appearance descriptor (it is calculated for each test box type) and $\|r_j\| = 1$. To ensure that the algorithm can still track the target after prolonged occlusion, the descriptors of the newest 100 frames on each trajectory are saved in R_i, i.e., R_i is the appearance feature vector set. When the cosine distance $d_2(i, j)$ is smaller than the training threshold of convolutional neural network, the association is considered to be successful.

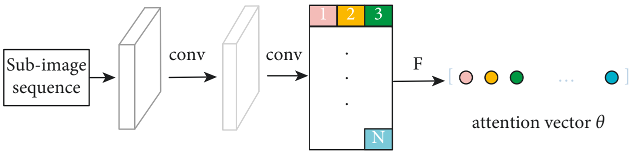

Figure 3: Structure of attention module.

The final decision information $C_{i,j}$ is obtained by (6) and adjusted by the superparameter λ. The smaller $C_{i,j}$, the greater the correlation between the detecting target and the tracking target. $C_{i,j}$ has a good effect on short-term prediction and matching, while the appearance feature can more efficiently measure the matching degree for long-lost tracks, which improves the robustness of the algorithm against target loss and occlusion.

2.3. Fall Judgment. According to the pedestrian detection and tracking results, the final fall events judgment is a binary classification problem. When a pedestrian is standing, the ground truth aspect ratio recognized is less than or equal to 0.4; when a pedestrian falls, the aspect ratio increases to 0.7~1.2. Meanwhile, the deflection angle is lower than a preset value (such as 37°), and the instantaneous acceleration in vertical direction is increased, which is significantly greater than that of squatting and bending. In this paper, the aspect ratio, deflection angle, and vertical instantaneous acceleration of the bounding box are comprehensively considered for the final fall judgment. These three factors not only have their own independence and meet the conditions of comprehensive judgment but also can avoid the higher dimension of feature vector space, overcomplex classifier, and poor real-time performance caused by excessive selection of feature vectors.

According to the tracking results, the authors can obtain the length H, width W, upper left point (x_L, y_L), and lower right point (x_R, y_R) of the bounding box in each frame. Then, the aspect ratio ρ is $\rho = W/H$. The bounding box (x_P, y_P) is

$$\begin{cases} x_P = (x_L + x_R)/2 \\ y_P = (y_L + y_R)/2 \end{cases}. \tag{9}$$

The deflection angle β of the bounding box is calculated by

$$\beta = \arctan \frac{y_P - y_L}{x_P - x_L}. \tag{10}$$

The authors denote M_i and M_{i+1} as two adjacent frames, respectively. The centroids of M_i and M_{i+1} are $(x_{\mathrm{Pi}}, y_{\mathrm{Pi}})$ and $(x_{\mathrm{Pi+1}}, y_{\mathrm{Pi+1}})$, respectively, and then the vertical velocity of the target in M_{i+1} is obtained by

$$v_{i+1} = \frac{\left| y_{Pi+1} - y_{\mathrm{Pi}} \right|}{t}, \tag{11}$$

where t is the time interval between M_i and M_{i+1}. Then, the vertical instantaneous acceleration a_{i+1} is

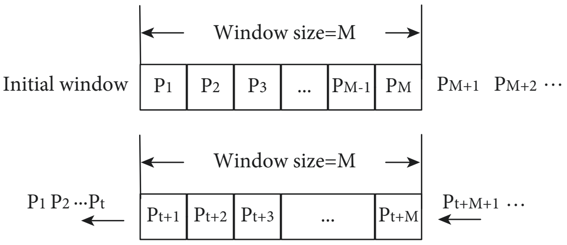

Figure 4: The sliding window diagram.

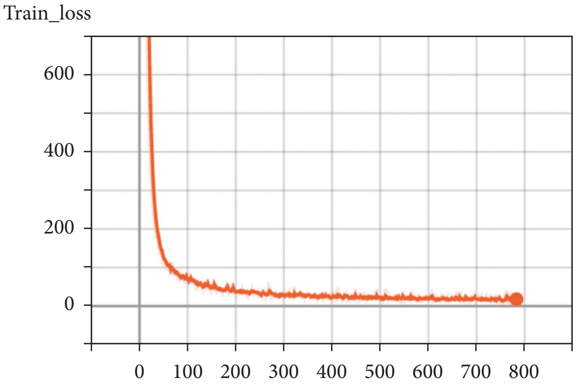

Figure 5: Loss curve.

$$a_{i+1} = \frac{v_{i+1} - v_i}{t}. \tag{12}$$

Because the traditional fall judgment algorithm only considers the factors of a single frame and the fall behavior has time continuity, the authors use the sliding window to obtain the variation of the three factors in continuous frames. As shown in Figure 4, the factors of the first frame are stored in a fixed size sliding window. As time goes on, the factors of subsequent frames continue to enter the container. After the container is filled, the newly entered factors are added at the end of the sliding window, and the leftmost data are removed.

Figure 4 shows that the size of the sliding window is M. For video frame sequence $\{P_1, P_2, \ldots, P_t, \ldots\}$, the initial window contains $\{P_1, P_2, \ldots, P_M\}$, where P_i stores the features obtained from moving targets, including human aspect ratio, deflection angle, and vertical instantaneous acceleration. After t frames, the content of the sliding window is $\{P_{t+1}, P_{t+2}, \ldots, P_{t+M}\}$. In this paper, the window size is set according to the fall behavior period. Since the fall period is about 0.5~0.8 seconds and the experimental video frame rate is 20 fps, the empirical value of window size is 15.

According to the feature information in the sliding window, a support vector machine (SVM) classifier [26] is constructed for fall detection training to determine whether a pedestrian has fallen. The training process inputs a large number of fall sample feature data and non-fall sample feature data into the SVM module to train a fall classifier by training these samples. Since extracting feature vector which

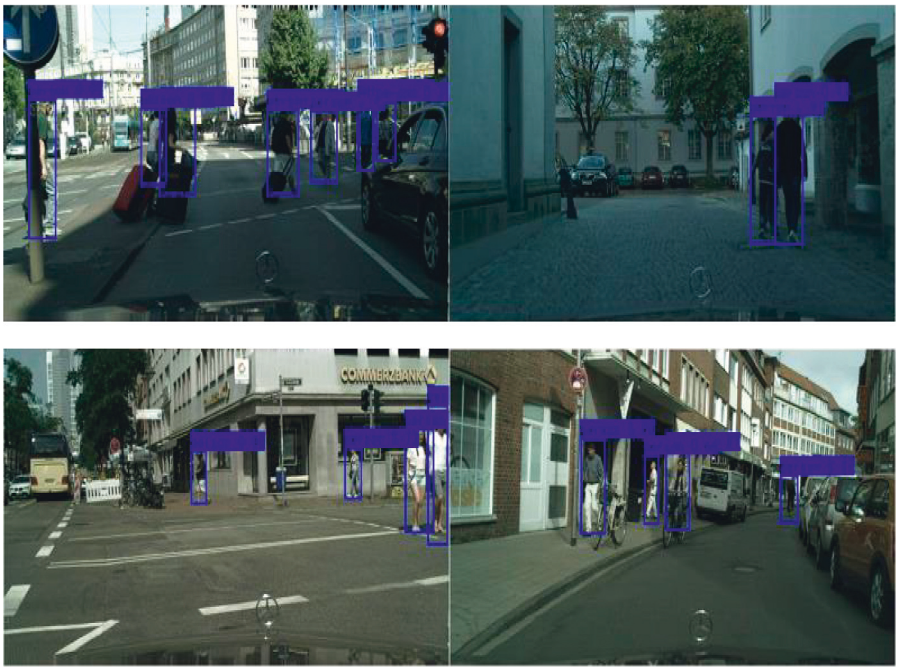

Figure 6: Detection results on the CityPersons dataset.

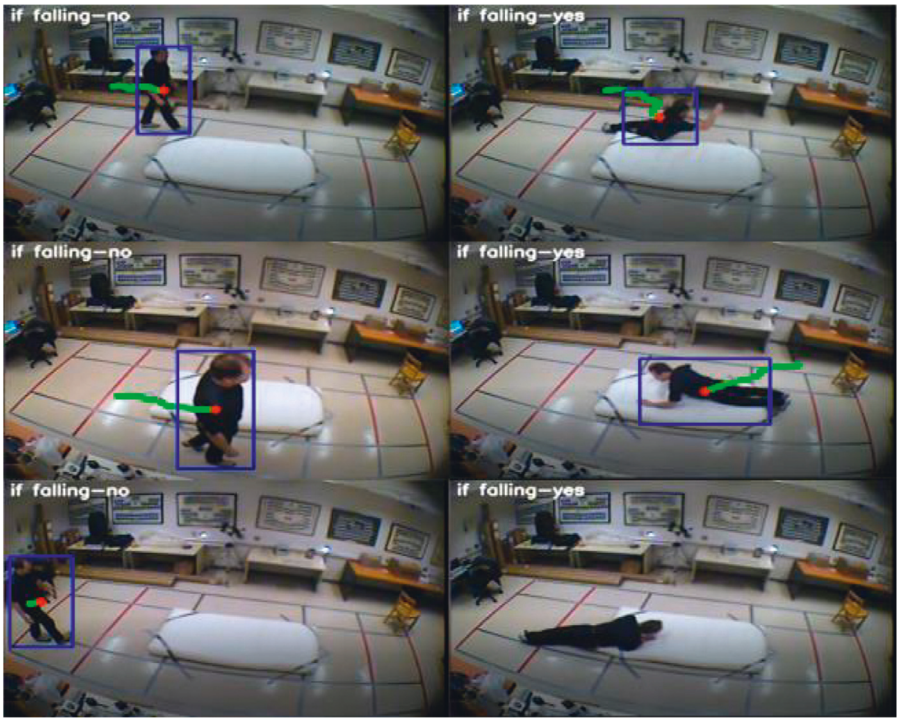

Figure 7: Fall event detection results on the Montreal fall dataset.

Table 1: Performance comparison conducted on the CityPersons dataset (%).

Algorithms	Average precision	mAP	fps
YOLOv1 [8]	96.33	76.13	67
YOLOv3 [24]	97.1	86.54	48
SSD [9]	97.27	78.37	41
Proposed	98.55	93.15	45

Table 2: Performance comparison conducted on the Montreal fall dataset (%).

Algorithms	Average precision
YOLOv1 [8]	96.33
YOLOv3 [24]	97.1
SSD [9]	97.27
Proposed	98.55

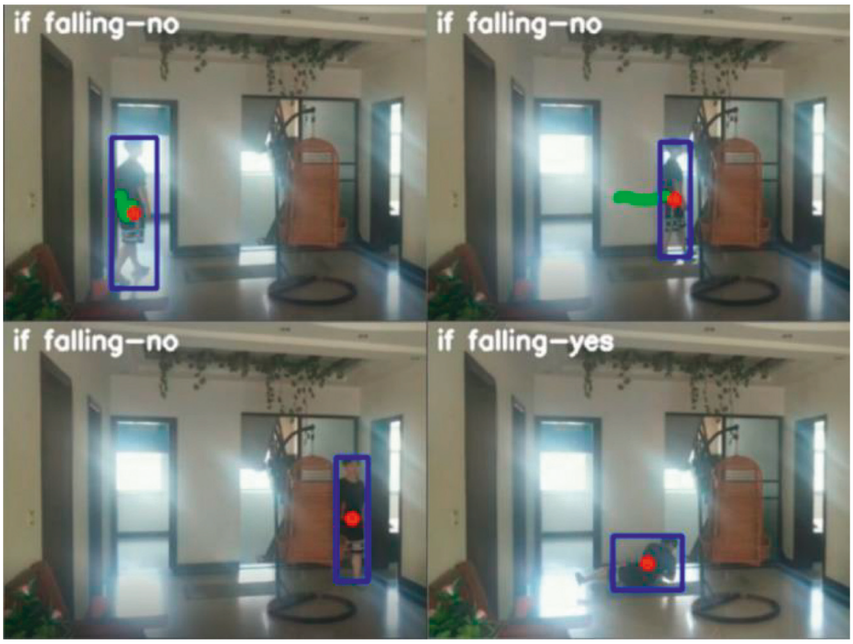

Figure 8: Test results of illumination change on self-built dataset in complex environments.

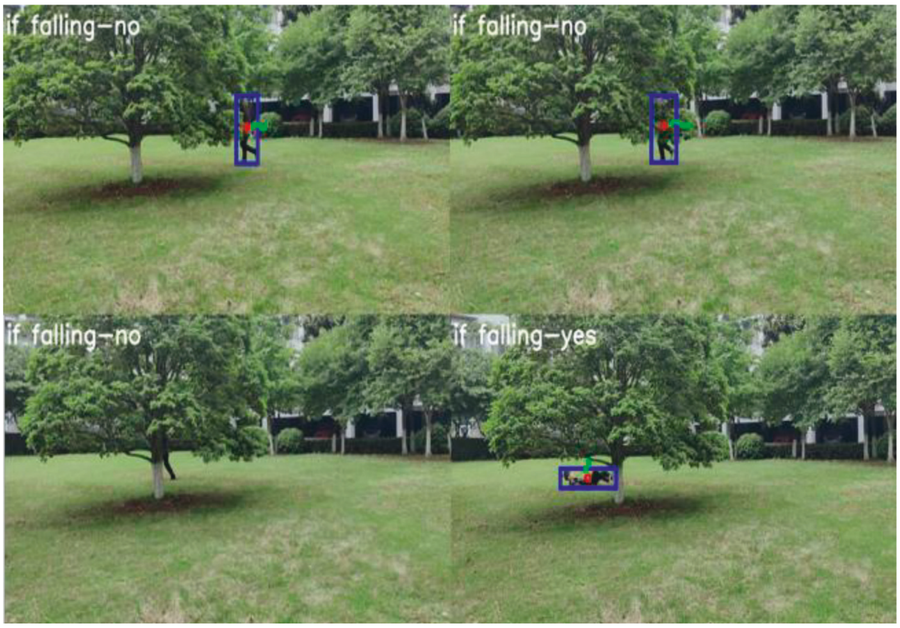

Figure 9: Results of outdoor occlusions.

is integrated of multiple features is a linear inseparable problem, the authors adopt the Gaussian kernel function to project the feature information into a high-dimensional space as

$$K(x,z) = e^{-\gamma \|x-z\|_2^2}, \tag{13}$$

where γ is a superparameter and $\gamma > 0$. It can be seen from, (13) that only a few parameters need to be adjusted.

3. Experiment

3.1. Dataset and Implementation. The experiments are conducted on the CityPersons [27] and the Montreal fall [28] datasets. The two datasets are built for supporting fall event detection study, and they are challenging for pedestrian detection. The CityPersons dataset is constructed based on Cityscapes dataset, which includes many video sequences, including 2975 image training sets and 1525 image test sets, with a resolution of 2048 × 1024. Among them, the Montreal

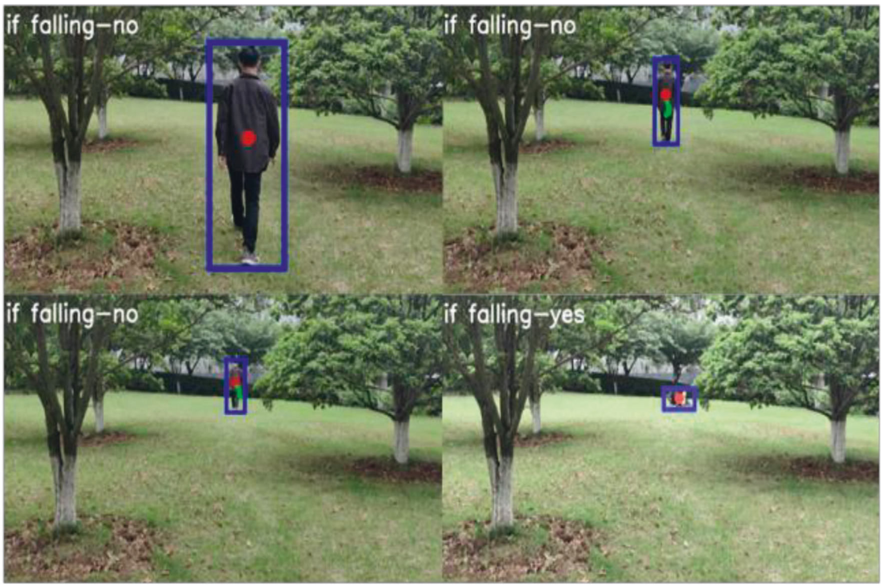

Figure 10: Results of outdoor scale changes.

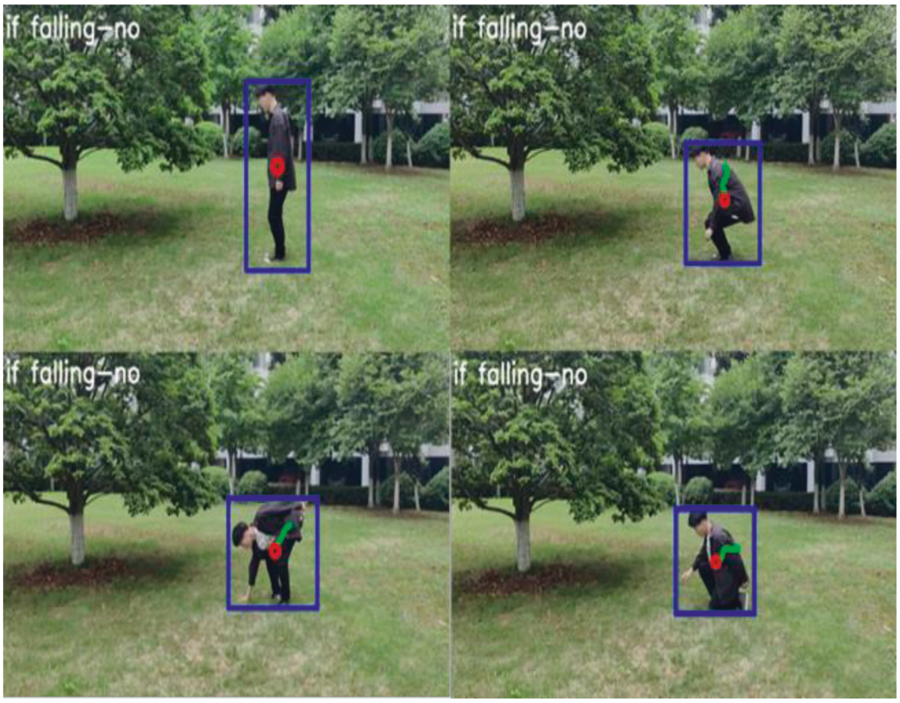

Figure 11: Results of interference experiment.

autumn dataset consists of 22 autumn events and 2 mixed events, which are recorded synchronously in the simulated daily life scenes. These videos are multi-view synchronized and can be used alone as a typical autumn event dataset or as a dataset for 3D scene reconstruction.

To diminish the effect of light illumination and scale transformation, the authors extend the test dataset through randomly changing the illumination, image size, angle, and so on. Accordingly, the original label box is adjusted, and the training set is extended.

The authors use the precision and recall rate, and the calculation formulas are described as follows:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \end{aligned} \tag{14}$$

where TP is true positive, FP is false positive, and FN is false negative, respectively.

In the experiment of this article, the YOLOv3 network is pretrained on the ImageNet [24] dataset. The initial learning rate, batch size, and epoch are set to 0.001, 8, and 200, respectively. Input image resolution is 416 × 416. The variation of training loss with the number of iterative steps is shown in Figure 5. Obviously, the model converges with the increase of iterative steps, and it reaches a low of about 800 and then gradually tends to be stable.

3.2. Experimental Results. The authors verify the proposed detector on the CityPersons and Montreal fall video dataset. Also, the visualization of detection results is described in Figures 6 and 7, respectively. Figure 6 shows the pedestrian detection results on CityPersons dataset. In Figure 6, the challenge includes the occlusions caused by other pedestrian or road signs or cars and different lighting and scales. It can be seen that the proposed detector can efficiently overcome the partial pedestrian occlusions and has certain robustness to the light and dark environment and different scales of pedestrians.

Figure 7 describes several fall event detection results on the Montreal fall dataset. There are three typical fall behaviors in different ways and angles, and the proposed method can automatically mark the detected fall event at the top right of the image accurately.

Tables 1 and 2 describe the detection performance on the CityPersons dataset and the Montreal fall dataset, respectively. It can be seen from the tables that the method proposed by the authors is better than YOLOv1 [7], YOLOv3 [24], and SSD [8] and achieves good performance similar to the most advanced methods.

To evaluate the proposed method in handling illumination changes and scale changes, the authors conduct test experiments on the self-built dataset. Figures 8–11 show some typical test results of self-built dataset in real complex scenarios. Figure 8 displays the fall event detection results of the indoor scene with high light and partial occlusion. Figure 9 shows the results of outdoor scene with pedestrian occluded by trees. Figure 10 shows the detection results handling scale changes, and Figure 11 shows the results handling four interferences of pedestrian: standing, squatting, bending, and squatting. Figures 8–11 show that the method proposed by the authors can handle strong light interference and keep tracking occluded pedestrians. In addition, for pedestrians of different scales, the proposed method can effectively adjust the bounding box and improve the accuracy of fall judgment. Especially, the proposed fall judgment algorithm has low judgment error for squatting, bending, and other interferences.

4. Conclusions

In this paper, the authors propose an attention-guided neural network to detect pedestrian fall events in complex scenes. By introducing the attention module into the detection framework, the attention module can guide the detector to focus on the key feature sequences of pedestrians; therefore, the detection performance is improved. In addition, the authors use the sliding window for storing feature maps and SVM classifier to redetect fall events. The experiments on CityPersons dataset, Montreal fall dataset, and the self-built dataset show the efficiency of the proposed attention-guided detection method. Although the proposed method can keep tracking occluded pedestrians well, it is necessary to develop the feature representation algorithm of pedestrians' body parts and improve detection precision for heavy occlusions.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (NSFC) under grant no. 61703201, the Natural Science Foundation of Jiangsu Province under grant no. BK20170765, the Foundation of Nanjing Institute of Technology under grant no. CKJB201905, the Graduate Innovation Program of Jiangsu Province under grant no. SJCX21_0945, and the College Students' Science and Technology Innovation Fund of Jiangsu Province under grant no. 202011276039Y.

References

[1] . Lee and 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.

[2] Kumar, Acharya, and B. Sandeep, "Wearable sensor-based human fall detection wireless system [J]," *Wireless Communication Networks and Internet of Things*, vol. 493, pp. 217–234, 2018.

[3] P. Tsinganos and A. Skodras, "On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection [J]," *Sensors*, vol. 18, no. 2, pp. 592–607, 2018.

[4] T. Xu, Y. Zhou, and J. Zhu, "New advances and challenges of fall detection systems: a survey [J]," *Applied Sciences*, vol. 8, no. 3, p. 418, 2018.

[5] K. Ghada, M. Mohamed, and B. Ouiem, "Automatic fall detection using region-based convolutional neural network [J]," *International Journal of Injury Control and Safety Promotion*, vol. 27, no. 4, pp. 546–557, 2020.

[6] X. Wang and K. Jia, "Human Fall Detection Algorithm Based on YOLOv3," in *Proceedings of the 5th IEEE International Conference on Image, Vision and Computing (ICIVC)*, pp. 50–54, IEEE, Beijing, China, 10 July 2020.

[7] P. Adarsh and P. Rathi, M. Kumar, "YOLO v3-Tiny: object Detection and Recognition using one stage improved model," in *Proceedings of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 687–694, IEEE, Coimbatore, India, 6 March 2020.

[8] A. Kumar and S. Srivastava, "Object detection system based on convolution neural networks using single Shot multi-box detector [J]," *Procedia Computer Science*, vol. 171, pp. 2610–2617, 2020.

[9] L. Zhang, L. Lin, X. Liang, and K. He, "Is Faster R-CNN Doing Well for Pedestrian Detection," in *Proceedings of the Conference on Computer Vision (ECCV)*, pp. 443–457, Springer, Netherlands, 11 October 2016.

[10] R. Dinakaran, L. Zhang, and A. Bouridane, "Deep learning based pedestrian detection at distance in smart cities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–6, IEEE, San Francisco, CA, USA, 18 June 2018, https://www.researchgate.net/profile/Faria-Mehboob.

[11] C. Rui, Y. Tong, and R. Liang, "Real-time generic object tracking via recurrent regression network [J]," *IEICE Trans. on Information and systems*, no. 3, pp. 602–611, 2020.

[12] F. Qi, C. Gao, and W. Lan, "Spatio-temporal fall event detection in complex scenes using attention guided LSTM [J]," *Pattern Recognition Letters*, vol. 130, pp. 242–249, 2020, https://www.sciencedirect.com/science/article/abs/pii/S016786551830504Xhttps://www.sciencedirect.com/science/article/abs/pii/S016786551830504X.

[13] A. Pramanik, S. K. Pal, J. Maiti, and P. Mitra, "Granulated RCNN and multi-class deep SORT for multi-object detection and tracking [J]," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–11, 2021.

[14] T. Zou, S. Yang, and Y. Zhang, "Attention guided neural network models for occluded pedestrian detection [J]," *Pattern Recognition Letters*, vol. 131, pp. 91–97, 2020, https://www.sciencedirect.com/science/article/abs/pii/S0167865519303733.

[15] C. Zhou and J. Yuan, "Multi-label learning of part detectors for heavily occluded pedestrian detection," *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pp. 3506–3515, 2017.

[16] A. Fernández-Isabel, P. Peixoto, M. Isaac, C. Conde, and E. Cabello, "Combining dynamic finite state machines and text-based similarities to represent human behavior [J]," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 504–516, 2019.

[17] J. L. Chua, Y. C. Chang, and W. K. Lim, "A simple vision-based fall detection technique for indoor video surveillance [J]," *Signal, Image and Video Processing*, pp. 623–633, 2015.

[18] J. Wu, S. Qiu, Y. Kong, Y. Wankou, L. Senhadji, and S. Huazhong, "PCANet: an energy perspective [J]," *Neurocomputing*, vol. 313, pp. 271–287, 2018.

[19] M. Zhang, C. Gao, and Q. Li, "Action detection based on tracklets with the two-stream CNN [J]," *Multimedia Tools and Applications*, vol. 77, no. 3, pp. 3303–3316, 2018.

[20] R. Girshick, J. Donahue, and T. Darrell, "Rich feature hierarchies for accurate object detection and semantic segmentation [C]," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, 2014.

[21] Z. Yi, S. Yongliang, and Z. Jun, "An improved tiny-yolov3 pedestrian detection algorithm [J]," *Optik - International Journal for Light and Electron Optics*, vol. 183, pp. 17–23, 2019.

[22] F. Ahmad, N. Li, and M. Tahir, "An Improved D-CNN Based on YOLOv3 for Pedestrian Detection," in *Proceedings of the IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, pp. 405–409, IEEE, ECMC, Buffalo, 19 July 2019.

[23] E. Zadobrischi and M. Negru, "Pedestrian Detection Based on TensorFlow YOLOv3 Embedded in a Portable System Adaptable to vehicles," in *Proceedings of the 2020 International Conference On Development And Application Systems (DAS)*, pp. 21–26, IEEE, Suceava, Romania, 21 May 2020.

[24] J. Redmon and A. Farhadi, "An Incremental Improvement," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 459–453, IEEE Computer Society, Washington, 18 June 2018.

[25] A. Bewley, Z. Ge, and L. Ott, "Simple Online and Realtime Tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, IEEE, Anchorage, AK, USA, 25 September 2016.

[26] A. Iazzi, M. Rziza, and H. Thami, "edchine," in *Proceedings of the 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6, Springer, 21 March 2018.

[27] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A Diverse Dataset for Pedestrian detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3224, IEEE, Honolulu, 21 July 2017.

[28] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, *Multiple cameras fall dataset [M]. Technical report 1350*, DIRO - Universite de Montreal, Quebec, Canada, 2010.