Fall Detection for Surveillance Video Based on Deep Learning

Hongwei Liu, Jiasong Mu(✉), and Zhao Zhang

Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin 300387, China
mjiasong@aliyun.com

Abstract. Accidental falls constantly threaten the lives of the elderly, and failing to detect them in time after a fall can cause the person to miss the best time to rescue, causing severe injury or even death. This paper proposes a video-based fall detection method to solve this problem. This method first performs motion detection of inter-frame difference on the video, performs feature extraction through deep learning, and finally completes classification by support vector machine to determine whether a fall occurs. The main application scene of this method is the surveillance video of the elderly living alone. The experimental results show that the proposed fall detection method has high accuracy and recall rate and can complete the fall detection task.

Keywords: Fall detection · Deep learning · Support vector machine

1 Introduction

With the continuous improvement of medical levels and living conditions in recent years, humans are living longer, and the problem of population aging has become more significant. With the increase of age, the physical function, balance ability, and coordination ability of the elderly are declining [1], which increases the probability of falls in the elderly. Falls have become a significant cause of injury and even death in the elderly. However, timely treatment after a fall can significantly reduce the mortality rate caused by a fall, and the more timely treatment is obtained, the lower the risk.

The current fall recognition methods are divided into three categories: methods based on environmental sensors, methods based on wearable devices, and strategies based on computer vision [2]. The way based on computer vision is different from the first two types. It does not require sensors to collect various parameter information. It mainly contains images or videos of human daily behavior and fall behavior and analyzes the obtained data to identify falls.

Transformer [3] has been making good progress and breakthroughs in Natural Language Processing since its introduction. The Vit [4] model completely replaces the convolution with a Transformer in the image processing task, cuts the picture into several small blocks, compares them to words in sentences, and inputs them into the model.

Q. Liang et al. (Eds.): CSPS 2022, LNEE 874, pp. 123–129, 2023.
https://doi.org/10.1007/978-981-99-2362-5_16

TimeSformer [5] extends the Vit model to a video understanding model, which divides the input video data into time series of image blocks extracted from each frame and similarly obtains feature information. The video feature extraction in this paper is mainly based on the Timeformer model.

2 Fall Detection Algorithm

The fall detection algorithm flow is shown in Fig. 1, including motion detection, video feature extraction, and fall detection in three stages.

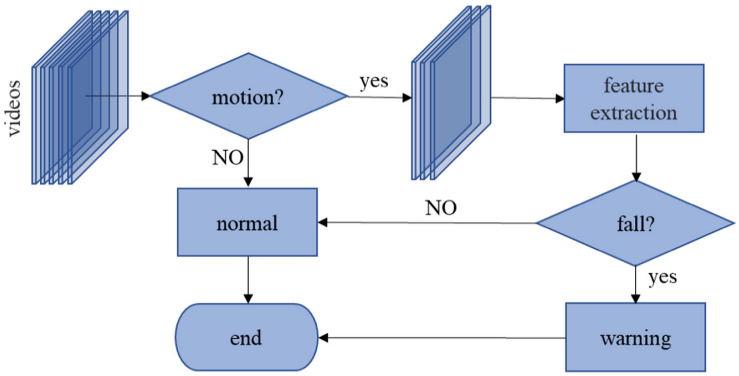


Fig. 1. Overall flow of fall detection algorithm

2.1 Motion Detection

This paper mainly uses the method of frame difference to detect motion. The difference between the two image frames can be obtained in the video by comparing the pixels. In the case of continuous frame illumination conditions, the pixel point of the difference image is not zero, which indicates that the target moves here. The primary process is as follows:

Firstly, read the video sequence frame, each frame image into a grayscale image, and differential calculation. Due to the slow movement of the elderly, the difference between adjacent frames is slight if the use of conventional two-frame difference is likely to cause misjudgment. Therefore, this paper uses the four-frame difference to take the average to reduce misjudgment. The specific method is to make the difference between the current frame and the last three frames, respectively, and take the mean value of the three different images to form the final difference image. The calculation formula is as follows:

$$D(x,y) = \frac{1}{3}(|F_t(x,y) - F_{t-1}(x,y)| + |F_t(x,y) - F_{t-2}(x,y)| + |F_t(x,y) - F_{t-3}(x,y)|) \quad (1)$$

Suppose that the image of the current time t is Ft, and the last three images are Ft − 1, Ft − 2, Ft − 3, and D representing the different images. The left of Fig. 2 is the image at time t, the middle of Fig. 2 is the image at time t − 3, and the right of Fig. 2 is the image after averaging the four-frame difference.

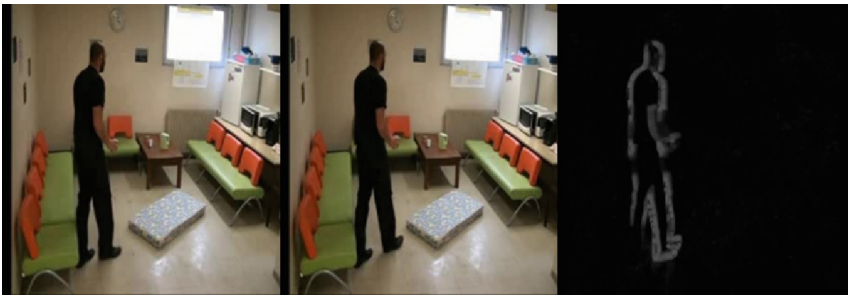


Fig. 2. Image at a different time and its mean difference image

Secondly, select an appropriate threshold T1, set the pixels greater than or equal to the threshold T1 as foreground pixels, set the pixels less than the threshold T1 as background pixels, and obtain the average gray value of foreground (avg1) and background pixels. The middle gray value of the pixel (avg2). d(x, y) is the value of the current differential image D at the (x, y) coordinate pixel point, and the calculation formula is as follows:

$$avg1 = \frac{1}{n} \sum_{x=1,y=1}^{n} d(x,y) \quad d(x,y) \geq T1 \tag{2}$$

$$avg2 = \frac{1}{m} \sum_{x=1,y=1}^{n} d(x,y) \quad d(x,y) < T1 \tag{3}$$

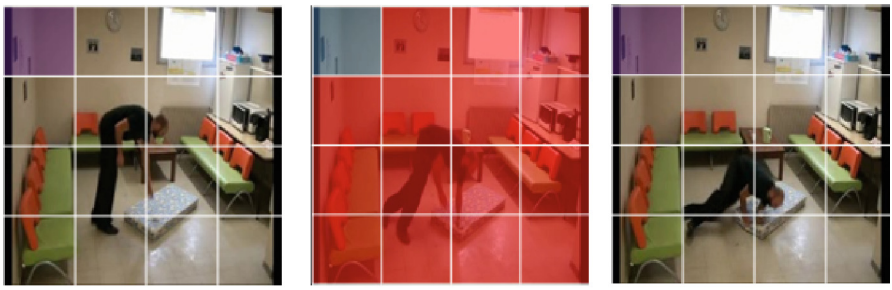
The threshold value T1 is determined by a threshold iteration method, which belongs to an adaptive determination method, and different images have different thresholds. It mainly uses multiple iterations to approximate the optimal threshold for the graph. Initialize a threshold T, calculate avg1, avg2, and Tn(Tn is the average of avg1 and avg2) under the current threshold, and use it as the initial threshold for the next round. If the difference between Tn and Tn-1 is slight, the iteration process ends, and Tn is determined as the threshold T1.

Finally, another threshold, T2, must be determined to calculate the ratio of the foreground pixel's average gray value to the background pixel's average gray value. If the percentage is greater than the threshold T2, the target moves; otherwise, the target does not move. Through the determination of the threshold, it can effectively reduce the misjudgment caused by random noise and improve the accuracy of motion detection.

2.2 Video Feature Extraction

Through the first stage of motion detection, the original video has been segmented into video clips containing motion information. In this stage, the video clips need input into

the TimeSformer, and the video features are extracted through the model to avoid complex and cumbersome manual feature extraction. The model first decomposes the input video to form a set of non-overlapping small image modules; to reduce the computational cost, time and space segmentation methods are adopted, and timely attention and spatial attention are used in turn [5]. Temporal attention is to compare the image modules in the same position of each frame image (the blue module in Fig. 3(b) and the purple module in Fig. 3(a) and Fig. 3(c)), and the input video has N frames. N temporal comparisons; spatial attention compares each imaging module with all other modules in the current frame (the blue module in Fig. 3(b) and all other red modules in Fig. 3(b)).



(a) **(b)** **(c)**

Fig. 3. **(a)** frame t − n. **(b)** frame n. **(c)** frame t + n

Extracting frames from the input video is adopted to extract features to improve the speed of extracting video features and reduce the waste of unnecessary computing resources. The extraction method is the average frame extraction method. For example, if F frames of video are extracted, one frame of image is extracted for every F/f frame of the video. Finally, the extracted video frame is input into the pre-trained TimeSformer model to obtain a high-dimensional feature vector.

2.3 Classification of Behavior States

After the video feature information is extracted from the second stage, a classifier is required to discriminate the video content. The support vector machine (SVM) [6] is a supervised learning algorithm. It maps the vectors of the data set into a high-dimensional feature space by the kernel function. Select a hyperplane in the area to classify the dataset so that vectors with different features are divided into both sides of the plane. If there are multiple such planes, the maximum distance between the two sides of the hyperplane is selected as the optimal plane.

SVM can effectively solve the small sample and binary classification problem, consistent with the fall detection to be achieved: the video data set selected in this paper is relatively small. The final classification can be regarded as a binary classification (fall and normal), so this paper chooses SVM as the classifier.

3 Experiments and Results Analysis

The experiment uses python3.8, OpenCV, PyTorch, and Sklearn in the PyCharm compiler environment. The experimental video size is 224 × 224, the frame rate is 24fps, the threshold T2 is 8, the penalty coefficient of the support vector machine is 0.9, and the kernel function is radial basis kernel function (RBF).

3.1 Data Sets and Processing

This paper selects two public datasets, the Le2i fall dataset [7], the Multiple cameras fall dataset [8], and the video collected in this experiment. Le2i dataset contains 191 videos from four different scenarios; the Multiple datasets include 24 scenes, each recorded from different angles using eight cameras; The collected data sets mainly include some common daily behaviors (falls, walking, etc.). The final dataset has a total of 700 videos (350 positive samples and 350 negative samples). The ratio of the training set and the test set is 4:1. Due to the uncertainty of the random partitioning of the dataset, the final model predictions may be questionable. Therefore, the cross-validation method combines the segmented data sets into different training sets and test sets. This paper is divided into ten groups.

3.2 Performance Index

The binary classification problem can divide the sample into four cases according to the combination of the actual category and the predicted category: true positive (TP, fall detection is fall); false positive (FP, non-fall detection is fall), true negative (TN, non-fall detection is non-fall) and false negative (FN, fall detection is non-fall). This paper selects the commonly used indicators of classification and evaluation: precision, recall rate, and curacy are used for performance measurement; the F1 measure is used to balance precision and recall. The calculation formula is as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{6}$$

$$F1 = \frac{2\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

3.3 Experimental Process and Results

In this 2.2, we need to extract the feature vector of video information by frame extraction. For video tasks, the video is generally extracted 8,16,24,32 frames. As the number of frames increases, the calculation speed of the model will be slower. This paper extracts the video data set into several standard frames and inputs it into the pre-trained TimeSformer model to extract high-dimensional feature vectors (768 dimensions). Each video corresponds to a feature vector to form a new vector data set. Finally, the support vector machine is used for training modeling and testing. Table 1 shows the classification results and indicators after extracting different frames.

Table 1. Test results of each frame number

Frame	Precision rate	Recall rate	Accuracy rate	F1
8 frames	91.96%	94.28%	93.01%	93.11%
16 frames	92.91%	95.43%	94.07%	94.15%
24 frames	94.23%	94.36%	94.29%	94.29%
32 frames	93.02%	91.42%	92.28%	92.21%

Although the overall effect of the model is the best when 24 frames of image data are extracted, 16 frames have a higher accuracy rate, and the accuracy rate is minimal because it should maintain a higher recall rate while meeting the accuracy rate. Because the two types of errors in detecting a fall as a fall and a fall as a fall are both misjudgments in statistics, but in practical applications, it is more desirable to have fewer occurrences of the latter category, and a higher recall rate is required. The reasoning speed of 16 frames is faster than that of 24 frames, so the method of extracting 16 frames is finally selected. In addition, training and testing experiments are performed only on the Le2i fall dataset and the Multiple cameras fall dataset. Table 2 shows the results, and Fig. 4 shows an example of overall model checking.

Table 2. Test results of each dataset

The data set	Precision rate	Recall rate	Accuracy rate	F1
Le2i	96.37%	98.52%	97.36%	97.44%
Multiple	91.89%	96.26%	93.88%	94.02%

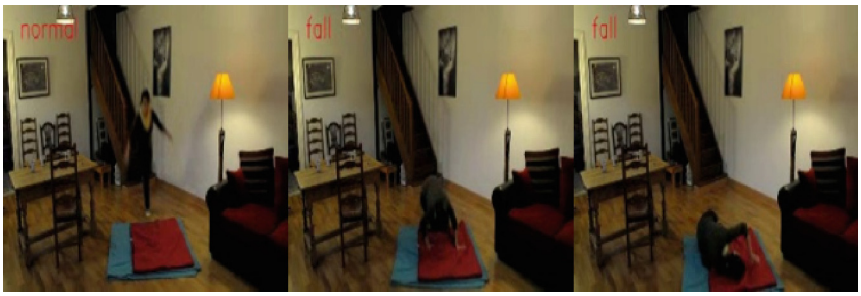


Fig. 4. Fall detection process

4 Conclusions

Fall detection is a way to monitor the lives of the elderly efficiently. This paper proposes a fall detection method based on video data input. Fall detection mainly involves motion detection, video feature extraction, and classification. The combination of neural network and support vector machine is used to analyze and discriminate the input video data containing motion. The experimental results show that the method can achieve better results. However, the model's generalization ability is affected due to the limited samples of the overall video data set. Therefore, it is necessary to construct a richer data set to improve the detection ability of the whole model.

References

1. Suqingqing, M., et al.: Analysis of fall status and influencing factors of the elderly in the senile apartment. J. Nurs. **37**(12), 16–18 (2022)
2. Ramachandran, A., Karuppiah, A., Gigantesco, A.: A survey on recent advances in wearable fall detection systems. BioMed Res. Int. (2020)
3. Vaswani, A., Shazier, N., Parmar, N., et al.: Attention Is All You Need (2017)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
5. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? (2021)
6. Zhihua, Z.: Machine learning. Tsinghua University Press, Beijing (2016)
7. Charfi, I., Miteran, J., et al.: Optimized Spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification. J. Electron. Image. **22**(4), 041106 (2013)
8. Adventinet, E., Rougier, C., et al.: Multiple cameras fall dataset, Technical report 1350, DIRO-Université de Montréal, July 2010