Vision-Based Fall Detection and Alarm System for Older Adults in the Family Environment

Fei Liu[1,2(✉)], Fengxu Zhou[1,2], Fei Zhang[1,2], and Wujing Cao[3]

[1] School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai 201209, China
liufei@sspu.edu.cn

[2] Smart Manufacturing Factory Laboratory, Shanghai Polytechnic University, Shanghai 201209, China

[3] Guangdong Provincial Key Lab of Robotics and Intelligent System, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 510663, China

Abstract. This study proposes an innovative fall detection and alarm system for the elderly in the family environment based on deep learning. The overall cost of hardware development is a camera and an edge device like a Raspberry PI or an old laptop that can detect and alert users to falls without touching the user's body. The development idea of the system is as follows: 1. Collect the pictures of falling and normal states under different conditions; 2. The improved lightweight SSD-Mobilenet object detection model is used to train the data set and select the optimal weight; 3. Optimal results are deployed on a Raspberry PI 4B device using a lightweight inference engine Paddle Lite. The mean Average Precision of the best model is 92.7%, and the detection speed can reach 14FPS (Frames Per Second) on the development board. When the camera detects that someone has fallen for 10 s, the compiled script sends an alert signal to the default guardian's email via the Mutt email program on Linux. The experimental results show that the fall detection system achieves satisfactory detection accuracy and comfort.

Keywords: Computer vision · Human fall detection · Object detection · Edge devices

1 Introduction

Nowadays, more and more people pay attention to home care. An essential part of daily care tasks is to detect falls in the elderly. The risk of falls is one of the most common problems faced by the elderly [1].

Fall is a significant cause of death in the elderly. It is especially dangerous for people who live alone, as it can take quite a long time before they receive help. Therefore, an effective fall detection system is essential for an elderly person, and in some cases can even save his life [2]. When an elderly person falls, the fall detection system detects abnormal behavior and sends an alert signal to some caregivers or the elderly person's family through modern communication.

H. Liu et al. (Eds.): ICIRA 2022, LNAI 13455, pp. 716–724, 2022.
https://doi.org/10.1007/978-3-031-13844-7_66

At present, researchers have proposed different methods to detect falls, which are mainly divided into two categories: non-visual detection method and visual detection method.

Non-visual detection method: Wearable sensor technology is the most commonly used type of commercial device for fall alarm products on the market, mainly in the form of pendants, belts, bracelets or watches [3, 4].

Visual detection method: Most commercial fall detection systems on the market are based on portable devices. Commercial devices based on computer vision are not common, but they look promising based on current vision-related technologies and literature.

In recent years, with the rapid development of pattern recognition technology, many researchers have applied this method to fall detection tasks [5]. Lu et al. developed a fall detection method based on 3D convolutional neural network, which only uses video motion data to train automatic feature extractor, thus avoiding the requirement of deep learning for large fall data sets [6]. Adrián Núñez-Marcos et al. proposed a vision-based solution that uses convolutional neural networks to determine whether a frame sequence contains a falling person. In order to model the video motion and make the system scene independent, they used optical flow images as the input of the network [7]. RicardoEspinosa et al. proposed a multi-camera fall detection system based on 2D convolutional neural network reasoning method. This method analyzes images in a fixed time window and uses optical flow method to extract features to obtain information about the relative motion between two continuous images [8].

Commercial devices for non-visual fall detection have been developed, but they largely require the elderly to wear sensor devices. Some older people, especially those with dementia, often forget to wear the device. Elderly people with dementia need intensive care to maintain independent living conditions.

In this study, we propose a visual fall detection and alarm system based on deep learning, which is mainly composed of an embedded computer and a camera. Firstly, the improved lightweight network model was used to extract the characteristics of subjects in the training set under normal and falling conditions, then the optimal model was deployed on the hardware platform of the Linux system (Raspberry PI, Nvidia Jetson series, notebook, etc.).

When the system detects that someone has fallen for 10 s or more, it sends an alert signal to the caregiver's mailbox via mutt on Linux using a preset script command. The device can be mounted on the wall or ceiling to monitor a room without human intervention. In addition, people monitored at home do not need to wear devices, and when detection occurs, an alarm email is sent to the preset caregiver. This method has the advantages of no impact on user comfort, low development cost and ideal detection rate.

2 Proposed Method

In this study, the object detection method was used to detect the occurrence of falls in the home environment. In order to make the detection model achieve the ideal detection effect on the low-cost hardware platform, an improved lightweight SSD (Single Shot MultiBox Detector) [9] algorithm was adopted to perform this task.

2.1 Overall Structure of Fall Detection and Alarm System

Figure 1 is the overall flow diagram of the fall detection and alarm system we developed.

First, deploy the optimal model to the terminal equipment (raspberry pie, NVIDIA Jetson series, computer, etc.) and install the equipment in a fixed position.

The system detects whether there is a fall event in the room in real time through the camera. The results of the object detection model have two categories (person and fall). When the detection is normal, there is no feedback signal. When an abnormal state is detected and lasts for more than 10 s, the system will send an alarm signal to the preset caregiver mailbox through mutt to remind them to deal with the dangerous event as soon as possible.

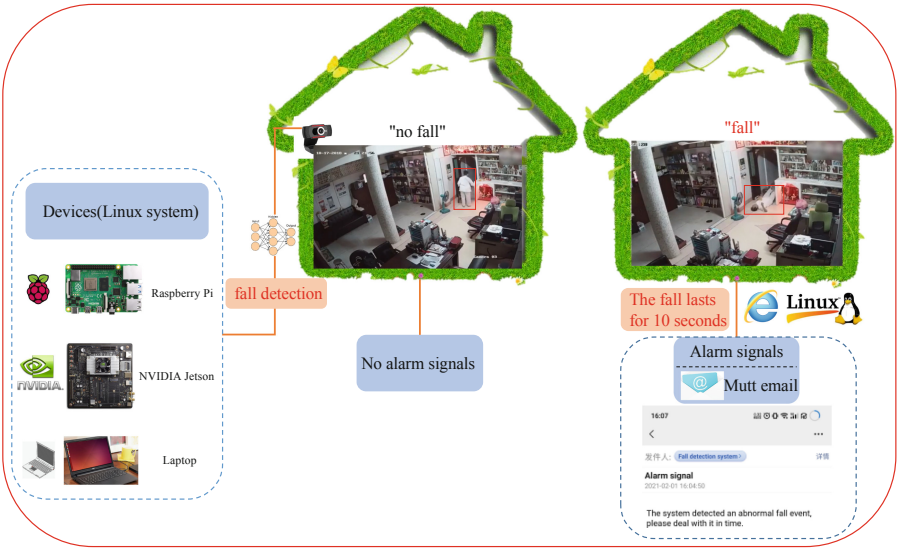

Fig. 1. Overall flow diagram of the system

2.2 Data Set

We used Python script to obtain 864 pictures of normal walking and 973 pictures of falling state from Baidu website. We also extracted 677 images from an open source fall data set. To enhance the robustness and generalization ability of the model, data enhancement methods were used to expand the data set. Specifically, not only common approaches such as rotation, cropping, and changing brightness and contrast but also generating new samples through the MixUp [10] algorithm to fuse positive and negative samples (Fig. 2).

Fig. 2. Data augmentation

2.3 Object Detection Algorithm

Algorithmic Network Structure. The object detection algorithm can judge whether there is a specified category in the detected image. Object detection algorithms can be divided into two main categories: one-step method and two-step method. The two-step method needs to be completed in two steps: regional proposal and detection. Its main advantage is high detection accuracy, such as R-CNN series. The one-step detection algorithm does not need to find candidate regions alone. Its main advantage is fast precision detection, such as SSD and YOLO series.

SSD is classified as a one-stage object detection method, which uses multiple frames to predict the object. Compared with the Faster R-CNN [11] algorithm, SSD can complete detection within one step, so the detection speed is faster. Candidate frames are obtained through convolutional neural network first, and then classification and regression are performed. Compared with YOLO algorithm [12], SSD algorithm overcomes the shortcomings of small object detection difficulty and inaccurate positioning.

The original SSD used VGG16 [13] backbone network as the basic model. VGG16 consists of 13 convolutional layers, 5 maximum pooling layers and 3 fully connected layers. The ReLU activation function is used after the hidden layer in the network.

The SSD algorithm makes some modifications to VGG16 as its backbone network. The main improvements include: removing fully connected layer 8, changing fully connected layer 6 and 7 to convolution layer, and performing secondary sampling for parameters of fully connected layer 6 and 7. The size of VGG pooling layer 5 was changed to 3×3, and the step size was changed to 1. To prevent overfitting, the dropout layer was removed. The modified VGG16 has excellent detection speed and accuracy. However, in low-configuration hardware platforms such as Raspberry PI, it still has a large amount of computation, and the real-time detection of the camera is very slow.

In order to reduce the computational burden of the model and improve the detection speed while maintaining the accuracy as much as possible, Mobilenetv1 model was used in this study instead of VGG16 to extract target features.

In 2017, Google Research released Mobilenetv1 [14] lightweight deep neural network (Fig. 3). Its main feature is that it uses depth-wise separable convolutional structure to replace the traditional convolutional mode. It is a convolutional neural network with

less computation and small volume, which is very suitable for deployment on platforms with limited computing power such as embedded or mobile terminals.

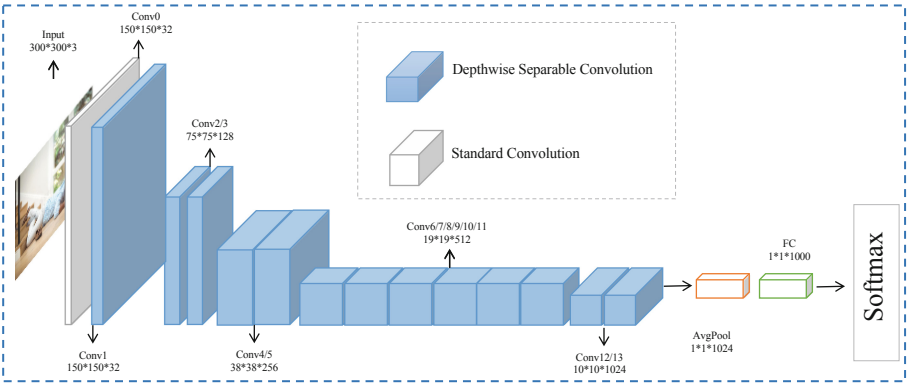

Fig. 3. Mobilenetv1 network architecture

In this study, some key modifications were made to the Mobilenetv1 network as the backbone network. Figure 4 shows the improved SSD-Mobilenetv1 network structure for fall detection. As can be seen from the basic network module of the algorithm, the input image in the new structure is uniformly set as 300 × 300, and the configuration from convolution 0 to convolution 13 is completely consistent with the Mobilenetv1 model, except that the global average pooling, fully connected layer and softmax layer of the last part of Mobilenetv1 are removed.

The SSD model used six different feature maps to obtain the features to be detected, with sizes of 19 × 19 × 512, 10 × 10 × 1024, 5 × 5 × 512, 3 × 3 × 256, 2 × 2 × 256, and 1 × 1 × 128, respectively. SSD-mobileNetV1 also uses six different feature maps, but the resolution of feature maps is only half that of SSD: 38 × 38 × 512, 19 × 19 ×

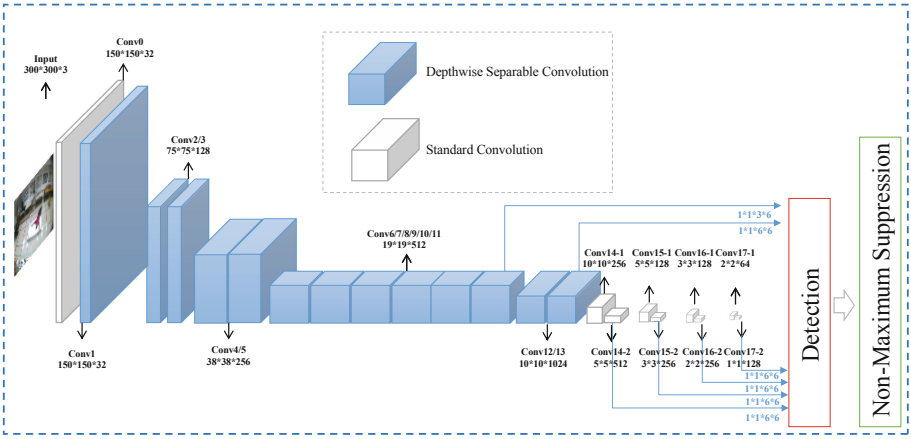

Fig. 4. SSD-Mobilenetv1 network architecture

1024, 10 × 10 × 512, 5 × 5 × 256, 3 × 3 × 256, and 1 × 1 × 128. In addition, in the path from feature graph to detection, the size of the convolution kernel used by SSD is 3 × 3, and the default number of boxes is 4, 6, 6, 6, 4, and 4 respectively. The size of the convolution kernel used by SSD-Mobilenetv1 is 1 × 1, and the default number of boxes is 3, 6, 6, 6, 6, and 6 respectively.

2.4 Loss Function of the Model

Model training is the process of reducing the error between predicted value and real value. The total target loss function of the improved SSD-Mobilenet algorithm used in this study is the weighted sum of position and classification loss, as shown below:

$$L(x, c, l, g) = \frac{1}{N}(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \tag{1}$$

$$L_{\text{conf}}(x, c) = -\sum_{i \in p_{os}}^{N} x_{ij}^{p} \lg \hat{c}_i^p - \sum_{i \in n_{eg}} \lg \hat{c}_i^0 \tag{2}$$

$$\hat{c}_i^p = \frac{\exp c_i^p}{\sum_p \exp c_i^p} \tag{3}$$

N is the number of matching boxes, l and g are the coordinates of predicted and real borders respectively, c represents the confidence of the softmax function on the target category, x is the matching mark between the predicted frame and the real frame, S_{L1} is the smooth L1 loss between forecast and true position, α is the weight coefficient, L_{loc} is the position loss, and L_{conf} is the classification loss, p_{os} and n_{eg} are the set of positive and negative samples respectively, b_{ox} represents the central coordinate, width and height of the prediction box.

3 Experiment

3.1 Model Training

The experimental training environment is shown in Table 1. Table 2 shows the initialization parameters of the improved SSD-Mobilenet network.

Table 1. Training environment.

Name	Model
CPU	Intel(R) Core i9-9900k (32 GB)
GPU	Nvidia Tesla V100 (16 GB)
Operating system	Ubuntu 16.04
Development language	Python 3.6
Deep learning framework	PaddlePaddle

Table 2. Initialization parameters of network.

Input size	Batch size	Learning rate	Num_workers	Iteration steps
300 × 300	32	0.001	8	30,000

3.2 Model Deployment and Test

We deployed the best model to the Raspberry Pi-4B. In order to ensure that the model can obtain excellent detection speed and accuracy, we used Paddle-Lite to compress the model to achieve the purpose of acceleration. The mean Average Precision of the best model is 92.7%. The accelerated model can reach the detection speed of 14FPS on this development board.

Fig. 5. Test results

In the alarm system, when the camera detects that the target is in a falling state and lasts for more than 10 s, Mutt tool will send an alarm message to the preset caregiver mailbox. Mutt is a text-based email client based on Linux system.

Figure 5 is the results of the test and alarm, which show that this method has good detection accuracy. This fall detection and alarm system can reduce the serious consequences of accidental falls to a certain extent.

4 Conclusion

The experimental results show that the method used in this paper has the advantages of comfortable use, ideal accuracy and low price. However, during the test, we found that there are occasional false detection situations, such as sometimes squatting will be judged as falling. In the next work, we will further expand the data set to classify various situations, and continue to optimize the network structure to improve the detection accuracy of the model. We will also try to apply the behavior detection model to fall detection task.

Acknowledgment. Research supported by follows: 1. Research Foundation of Shanghai Polytechnic University under grant EGD22QD01; 2. Guangdong Basic and Applied Basic Research Foundation under grant 2021A1515011699.

References

1. Ren, L., Peng, Y.: Research of fall detection and fall prevention technologies: a systematic review. IEEE Access **7**, 77702–77722 (2019)
2. Xu, T., Zhou, Y., Zhu, J.: New advances and challenges of fall detection systems: a survey. Appl. Sci. **8**(3), 418 (2018)
3. Santos, G.L., Endo, P.T., Monteiro, K.H.C., et al.: Accelerometer-based human fall detection using convolutional neural networks. Sensors **19**(7), 1644 (2019)
4. Mauldin, T.R., Canby, M.E., Metsis, V., et al.: SmartFall: a smartwatch-based fall detection system using deep learning. Sensors **18**(10), 3363 (2018)
5. De Miguel, K., Brunete, A., Hernando, M., et al.: Home camera-based fall detection system for the elderly. Sensors **17**(12), 2864 (2017)
6. Lu, N., Wu, Y., Feng, L., et al.: Deep learning for fall detection: three-dimensional CNN combined with LSTM on video kinematic data. IEEE J. Biomed. Health Inform. **23**(1), 314–323 (2018)
7. Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Vision-based fall detection with convolutional neural networks. Wirel. Commun. Mob. Comput. **2017**, 1–16 (2017)
8. Espinosa, R., Ponce, H., Gutiérrez, S., et al.: A vision-based approach for fall detection using multiple cameras and convolutional neural networks: a case study using the UP-Fall detection dataset. Comput. Biol. Med. **115**, 103520 (2019)
9. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
10. Zhang, H., Cisse, M., Dauphin, Y.N., Lopezpaz, D.: mixup: beyond empirical risk minimization. arXiv: Learning (2017)

11. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
12. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
14. Howard, A.G., Zhu, M., Chen, B., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)