IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 11, 2019, accepted November 26, 2019, date of publication January 13, 2020, date of current version January 28, 2020.

*Digital Object Identifier 10.1109/ACCESS.2019.2962778*

# A Two-Stream Approach to Fall Detection With MobileVGG

**QING HAN[1], HAOYU ZHAO[1], WEIDONG MIN[2,3], (Member, IEEE), HAO CUI[1], XIANG ZHOU[2], KE ZUO[2], AND RUIKANG LIU[1]**

[1]School of Information Engineering, Nanchang University, Nanchang 330031, China
[2]School of Software, Nanchang University, Nanchang 330047, China
[3]Jiangxi Key Laboratory of Smart City, Nanchang 330047, China

Corresponding author: Weidong Min (minweidong@ncu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61762061, in part by the Natural Science Foundation of Jiangxi Province, China, under Grant 20161ACB20004, and in part by the Jiangxi Key Laboratory of Smart City under Grant 20192BCD40002.

**ABSTRACT** The existing deep learning methods for human fall detection have difficulties to distinguish falls from similar daily activities such as lying down because of not using the 3D network. Meanwhile, they are not suitable for mobile devices because they are heavyweight methods and consume a large number of memories. In order to alleviate these problems, a two-stream approach to fall detection with the MobileVGG is proposed in this paper. One stream is based on the motion characteristics of the human body for detection of falls, while the other is an improved lightweight VGG network, named the MobileVGG, put forward in the paper. The MobileVGG is constructed as a lightweight network model through replacing the traditional convolution with a simplified and efficient combination of point convolution, depth convolution and point convolution. The residual connection between layers is designed to overcome the gradient disappeared and the obstruction of gradient reflux in the deep model. The experimental results show that the proposed two-stream lightweight fall classification model outperforms the existing methods in distinguishing falls from similar daily activities such as lying and reducing the occupied memory. Therefore, it is suitable for mobile devices.

**INDEX TERMS** Deep learning, fall detection, motion characteristics, the two-stream model, the MobileVGG.

## I. INTRODUCTION

A fall refers to a person's sudden, involuntary, unintended position change, such as falling onto the ground or onto a lower place. Falling is the number one killer of the accidental injury of the aged [1]–[3]. With the development of the aging society, an increasing number of old people live alone and are not taken care of. Falls become a severe issue in the care of the elderly. As many old people have chronic diseases, even minor falls may threaten their health and life. Some old people luckily survived from falls but rely on others' care in daily life and need medical aids for walking. Therefore, the development of automatic fall detection has become an urgent need for protecting vulnerable people, especially the old. At the same time, it has become a hot research topic.

The associate editor coordinating the review of this manuscript and approving it for publication was Hualong Yu.

The existing fall detection methods are divided into the fall detection method based on the auxiliary equipment and the computer vision. The fall detection based on auxiliary equipment usually depends on some wearable devices, in which there are a variety of sensors. The fall detection methods based on the computer vision are divided into the method using traditional geometric features and that based on deep learning methods. As for the deep learning method, the traditional Convolutional Neural Network (CNN) models such as AlexNet, VGGNet as well as other advanced networks are adopted [4]. Although the existing deep learning methods have obtained great achievements in human fall detection. But it is difficult for them to distinguish falls from similar daily activities such as lying down without the 3D Network. Meanwhile, they are not suitable for mobile devices because they are heavyweight methods and consume a large number of memories [5]. While the mobile devices are extensively

used in many scenarios including fall detection using mobile devices and the development of more economical lightweight fall detection devices. Apparently, the above issues limit the application of the existing fall detection methods.

In order to alleviate these problems, a two-stream approach to fall detection with the MobileVGG is proposed in this paper. One stream relies on the motion characteristics of the human body, while the other stream is an improved lightweight VGG network, named the MobileVGG, introduced in the paper. When the MobileVGG is improved and designed, the structures of the MobileNet [6] and the ResNet [7] have been analyzed and researched. To be specific, the MobileNet uses depth detachable convolution to construct the lightweight deep neural network. The ResNet adopts residual connection structure to accelerate the training of the neural network and enhance the accuracy of classification. The proposed MobileVGG is improved on the basis of VGG16 with the structure adjusted. It is constructed in a lightweight model and is featured with efficient fall classification capability.

The main contributions of this paper are as follows:

a) A two-stream approach to fall detection with the MobileVGG is proposed. It outperforms the existing methods in distinguishing falls from similar daily activities such as lying and reducing the occupied memory. In this case, it is suitable for mobile devices.

b) The MobileVGG is put forward to accelerate the training process. It is constructed as a lightweight network model through replacing the traditional convolution with a simplified and efficient combination of point convolution and depth convolution. The residual connection structure between layers is designed to overcome the disappearance of the shallow parameter gradient and the obstruction of gradient reflux in the deep model.

The rest of this paper is organized as follows. In section II, the related work is discussed. Section III describes the details of the proposed two-stream fall detection method. In Section IV, the proposed lightweight MobileVGG is elaborated. Section V involves experiment results, while Section VI gives conclusions.

## II. RELATED WORK

In this section, the related work of fall detection will be reviewed from three aspects: method based on wearable devices, method based on computer vision and a summarization.

### A. METHOD BASED ON WEARABLE DEVICES

There has been a fair amount of research on fall detection. Early methods rely on devices, such as wearable devices or environmental sensors [8]–[11]. This kind of method usually uses wearable devices to track the motion of the human body and record the parameter characteristics of some motions. Pierleoni *et al.* [12] extracted human motion parameters with the help of the tri-axial accelerometer, the gyroscope and the magnetometer. A fall detection system embedded in sensing equipment is designed. Ozcan *et al.* [13] placed a sensor camera on the human body to determine whether the fall occurred or not by comparing the changes of the angle of view. Ejupi *et al.* [14] designed a wearable device based on the wavelet algorithm to detect and evaluate the motion process from sitting-to-station and the risk of old people falling. Ando *et al.* [15] collected environmental information with the help of smart phones equipped with various embedded sensors, and finally achieved the purpose of detecting various daily behaviors. Mehmood *et al.* [16] proposed a novel fall detection technique with the wearable SHIMMER$^{TM}$ sensors, which identify the fall using Mahalanobis distance on real-time data. It is more robust than other conventional distance measure techniques, followed in existing fall detection systems. Kerdjidj *et al.* [17] put forward an efficient automatic fall detection system, which is also fit for the detection of different activities of daily living (ADL) and relies on a wearable shimmer device to transmit some inertial signals via a wireless connection to a computer.

First of all, the fall detection method based on auxiliary equipment needs to put a variety of sensors or other devices in a specified position, which makes this kind of method limited by the accuracy, the angle and other factors of the auxiliary equipment. Secondly, the pressure, vibration and other environmental characteristics used in this method are particularly sensitive to external factors, and the stability is not well guaranteed. So, this sort of method has poor anti-noise ability. Finally, wearable auxiliary equipment is bound to bring inconvenience to the user's life. It may cause more serious discomfort to the user.

### B. METHOD BASED ON COMPUTER VISION

Because of a series of problems in equipment assistance, people try to use the computer vision [18], [19] to cope with the issue of motion detection, including fall detection. The common approach is based on traditional machine learning such as SVM and traditional CNN models. Min *et al.* [20] proposed a fall detection method for indoor environments based on the Kinect sensor and analysis of three-dimensional skeleton joint information. Ravanbakhsh *et al.* [21] proposed a new metric-based method to measure local anomalies in the video by combining semantic information with low-level optical streams. Ullah *et al.* [22] put forward a novel action recognition method using the Convolution Neural Network and the Deep Bidirectional LSTM (DB-LSTM) network to process video data. Hou *et al.* [23] proposed an end-to-end depth network for the video action detection-pipeline convolution neural network (T-CNN). The architecture is a unified deep network, which can recognize and locate actions in line with 3D convolution features. Yan *et al.* [24] proposed an action detection system in office environment on the strength of the deep convolution neural network. After introducing the combination of partial order and RCNN, the scope of contextual cues is extended to the basic attitude components, which plays an important role in action recognition. Nait *et al.* [25] studied whether

deep learning methods from machine learning are suited to assess fall risk using automatically derived features from raw accelerometer data. They compared the performance of three deep learning model architectures, i.e. the CNN, the Long Short-Term Memory (LSTM) and a combination of these two (ConvLSTM), to each other and to a baseline model with biomechanical features on the same dataset. Solbach and Tsotsos [26] presented a new, non-invasive system for falling people detection. That approach uses merely stereo camera data for passively sensing the environment. A human fall detector uses a CNN based human pose estimator in combination with stereo data to reconstruct the human pose in 3D and estimate the ground plane in 3D.

Some people utilize the 3D network to solve this problem. Lu *et al.* [27] proposed a fall detection method based on 3D CNN. This method only uses video motion data to train the automatic feature extractor without the requirement of the large drop dataset of deep learning solution. Wang *et al.* [28] proposed the integration of visual attributes including detection, coding and the classification into multi-stream 3D CNN for action recognition in clipped videos, and proposed a visual attribute enhanced 3D CNN (A3D) framework. Ji *et al.* [29] proposed a new 3D CNN action recognition model, which performs the 3D CNN and extracts traits from spatial and temporal scales to capture motion information encoded in multiple adjacent frames. Lu *et al.* [30] proposed a fall detection method based on the 3D CNN. This method solely uses video motion learning data to train the automatic feature extractor, which avoids the requirement of deep learning solution for large datasets.

In addition to deep learning methods, people also endeavor to integrate multiple information for motion detection. Zhang *et al.* [31] proposed a two-stream real-time action recognition method using the motion vector instead of the optical stream. Tu *et al.* [32] proposed a multi-stream convolution neural network structure to identify human behavior. Besides, the method considers the regions related to human beings that contain most informative features as well. Gkioxari and Malik [33] used rich feature levels of the shape and motion clues to establish a motion model. The CNN was used to extract the space-time feature representation to construct a strong classifier, which obtained good results. Feichtenhofer *et al.* [34] studied many methods of the fusion of the ConvNet tower in space and time, and proposed a brand-new video behavior detection method for the space-time fusion of video fragments. Simonyan and Zisserman [35] proposed a two-stream network model combined with the space-time network. Multi-task learning is applied to two disparate action classification datasets, which proves that it can increase the amount of training data and improve the training effect. In addition to these, some people also use image shape features for fall detection. Min *et al.* [36] proposed an automatic human fall detection method using the Normalized Shape Aspect Ratio (NSAR). The method integrates the NSAR with the moving speed and direction information to robustly detect human fall, as well as being able to detect falls toward eight different directions for multiple humans.

### C. SUMMARIZATION

To summarize the above-mentioned contents, the existing motion behavior detection models based on conventional CNN have achieved good results in the accuracy. But the structure of these models has many problems, such as complexity, the large number of parameters and unsuitability for mobile devices. The network with high redundancy and difficulty to train will inevitably affect the efficiency of the model. Therefore, the lightweight classification network has become an inevitable trend. The lightweight network model not only lessens parameters of the model while maintaining the classification accuracy, but also reduces the proportion of the model and improves the efficiency of fall detection. Therefore, it is necessary to design a lightweight network classification model for fall detection. Apart from this, the existing deep learning methods for human fall detection have difficulties in distinguishing the fall from similar daily activities such as lying down without using the 3D network. Meanwhile, they are not suitable for mobile devices because they are heavyweight methods and consume large number of memories. Therefore, in order to alleviate the above problems, a two-stream approach to fall detection with the MobileVGG is proposed in this paper.

## III. THE PROPOSED TWO-STREAM FALL DETECTION METHOD

### A. AN OVERVIEW OF THE PROPOSED METHOD

The goal of this work is to establish a two-stream fall detection model based on motion characteristics and lightweight networks. The general framework of the proposed method is shown in Fig. 1. The proposed approach to fall detection with the MobileVGG consists of two streams. One stream is based on the motion characteristics of the human body for detection of falls. The other stream is an improved lightweight VGG network, named MobileVGG. Firstly, the video is decomposed frame by frame. Secondly, each video frame is sent to the motion characteristics and the proposed MobileVGG network at the same time. For the motion characteristics stream, it detects whether there is a fall behavior. If the fall behavior is detected, then the MobileVGG begin to work further to confirm whether it is a fall behavior. If the fall behavior is not detected, MobileVGG will not further detection.

Using the motion characteristics stream can distinguish the fall from similar daily activities such as lying down, while avoiding the use of more intricate 3D Networks.

### B. STREAM ONE: FALL DETECTION USING MOTION CHARACTERISTICS

The video frame sequence captured by the camera has a strong continuity. If there is no moving target in the scene, few differences between adjacent frames can be seen. Once the moving target appears, the moving parts can be extracted

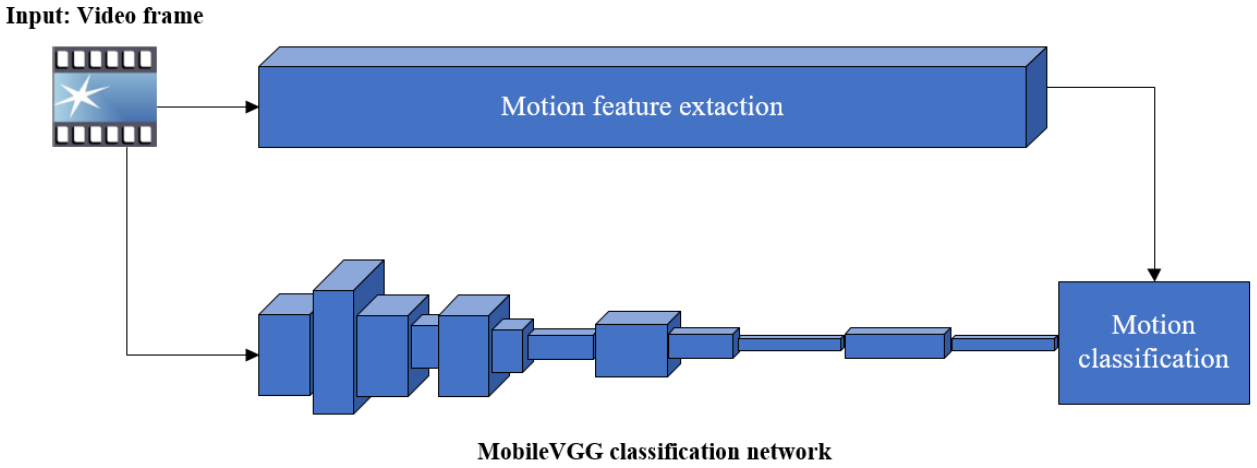



**FIGURE 1.** Fall detection using motion characteristics and fall detection with MobileVGG.

by the obvious change between adjacent frames. Especially the video taken from the indoor surveillance camera is very suitable for such analysis scenes. Because indoor scenes include less interferences such as moving walkers and be much quieter. In the video obtained by indoor camera, usually only the human movement will cause the change between adjacent frames. In this paper, the frame difference method is used to extract the motion features of the moving human body.

Note that the images corresponding to the frame $n$ and the frame *n-1* in the video sequence are $I_n$ and $I_{n-1}$. Then the difference image $D_n$ can be got as in (1):

$$D_n = |I_n - I_{n-1}| \tag{1}$$

The difference of corresponding position pixels was calculated by using (1). By setting the threshold $T$, the obviously changed pixels can be retained and other pixels will not be noticed. Then the position of the moving target in the difference image can be obtained which is a rectangular box of the human body outline in the target region. Assuming that the image of the moving object is $R$, then $R$ can be represented as in (2):

$$R = \begin{cases} 255, & D_n > T \\ 0, & ELSE \end{cases} \tag{2}$$

Obviously, when abnormal behavior occurs, the physical state of the human body will be break away from the stable state. The horizontal speed and vertical speed of the human body will both immediately change especially when the human body falls.

Hence, $R$ is used to calculate the horizontal and vertical velocities of the human body as the first stream of fall detection. The second classification of the fall of the classification model is carried out immediately, as another stream, When the speed of the human movement appears abnormal.

In the image $R$, the center of the human body $\underset{H_{c_i}}{\longrightarrow}$, the width of the human body detection frame $H_w$ and the height of the human body detection frame $H_h$ are represented as $Persion_v = \left\{ \left( \underset{H_{c_i}}{\longrightarrow}, H_{w_i}, H_{h_i} \right) | i \in N \right\}$. The vertical velocity $V_h$ was defined as in (3). It is the ratio of the height difference and the time $T$ between two frames. The horizontal velocity $V_{\mathrm{w}}$ as in (4) is the ratio of the width difference and the time between adjacent frames.

$$V_h = (H_{w_i} - H_{w_{i-1}})/T \tag{3}$$

$$V_w = (W_{h_i} - W_{h_{i-1}})/T \tag{4}$$

When the velocity values in both $V_h$ and $V_{\mathrm{w}}$ change, the values of $V_{h_i} - V_{h_{i-1}}$ and $V_{w_i} - W_{w_{i-1}}$ will be calculated. When the $V_{h_i} - V_{h_{i-1}}$ and $V_{w_i} - W_{w_{i-1}}$ values exceed the threshold, it signifies that the human behavior has changed greatly at this time. It can be predicted such a video frame as falling behavior since most of these scenes occur at the time of falling.

The sudden motion of the human body and other possible rapid behavior may also result the error of the fall detection. So, the MobileVGG classification stream was used to confirm whether it is a falling behavior when such video frame generates.

At the same time, the detection of the fall based on motion characteristics can effectively solve the problem that the CNN classification model cannot correctly distinguish the fall from similar daily activities like lying down. That's because lying down behavior in a static frame looks like a fall behavior. Such frames would not cause mutations in $\mathbf{V_h}$ and $\mathbf{V_w}$. So, it is not detected when the motion characteristics stream was used.

### C. STRRAM TWO: FALL DETECTION WITH THE MOBILEVGG

This work trained the MobileVGG network to complete the further fall detection. As shown in the Fig. 1, if the fall behavior is detected after the video frame passes through the

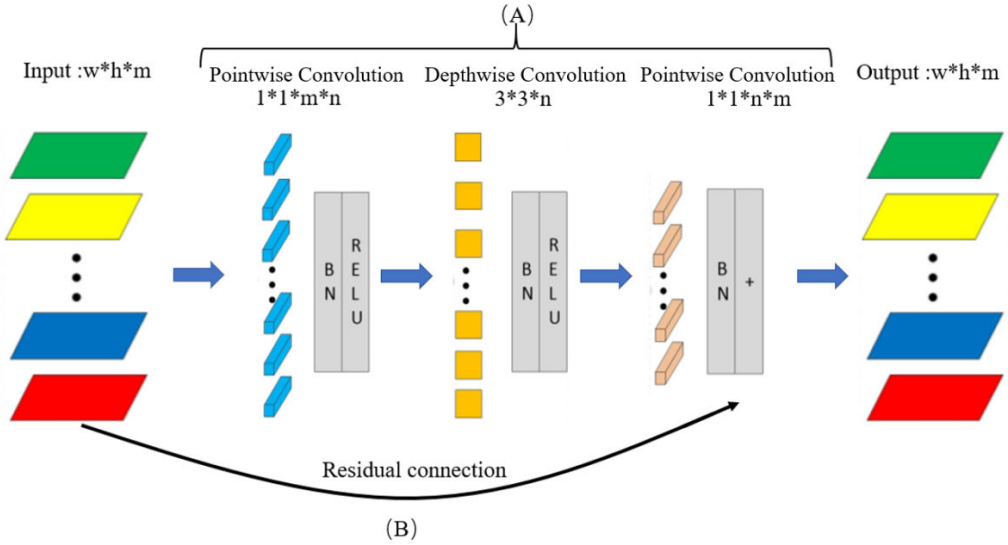



**FIGURE 2.** Proposed lightweight convolution neural network structure.

moving characteristics stream, the MobileVGG classification network will work to confirm that the fall behavior has indeed occurred in this video frame. The details of the proposed MobileVGG network are presented in section IV.

## IV. THE PROPOSED MOBILEVGG NETWORK

### A. AN OVERVIEW OF THE MOBILEVGG NETWORK

Fig. 2 provides an overview of the network structure. The MobileVGG is constructed as a lightweight network model through replacing the traditional convolution with a simplified and efficient combination of the point convolution and the deep convolution. The residual connection structure in the input and output of the specific layer is established to ensure that the network has faster convergence speed and higher accuracy.

### B. TRADITIONAL CONVOLUTION NEURAL NETWORK STRUCTURE

The lightweight network MobileVGG belongs to the classification network. Therefore, it can be started with a two-classification problem. Suppose each point represents an independent sample of the same distribution. $N_1$ denotes the number of categories $C_1$ and $N_2$ represents that of category $C_2$. The characteristic of each sample is $x$. The classification problem is to figure out the probability of confidence that the sample is an input feature. The probability $P$ is calculated as in (5):

$$P(C_i|x) = \frac{p(x|C_i)p(C_i)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)}, \quad (i = 1, 2) \tag{5}$$

where amount $P(C_1) = N_1/(N_1 + N_2)$, $P(C_2) = N_2/(N_1 + N_2)$. What people want to know is the confidence of $P(x|C_1)$ and $P(x|C_2)$ in the classification of each category as the judgment of the classification.

In order to solve the desired confidence levels $P(x|C_1)$ and $P(x|C_2)$. The hypothetical class obeys Gaussian distribution $P_{\mu_i,\sum_i}(x|C_i)$. The mean value is $\mu_i$, variance is $\sum_i$. Then the distribution function is calculated as in (6):

$$P_{\mu_i,\sum_i}(x \mid C_i) = \frac{1}{|\sum_i|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)^T \sum_i^{-1}(x-\mu_i)\right\} \tag{6}$$

Owing to the independent and identical distribution of each sample, the joint probability of class $i$ can be obtained as $P_{\mu_i,\sum_i}$ in (7):

$$P_{\mu_i,\sum_i} = P(x_1 \mid C_i)\, P(x_2 \mid C_i) \ldots\ldots P(x_k \mid C_i) \ldots\ldots \tag{7}$$

The unknown parameters mean $\mu_i$ and variance $\sum_i$ that need to be determined can be represented as in (8):

$$\mu_i^*, \sum\nolimits_i^* = arg \max_{\mu_i,\sum_i} P_{\mu_i,\sum_i} \tag{8}$$

Make $\frac{(\partial P(\mu_i,\sum_i))}{\partial \mu_i} = 0$ and $\frac{(\partial P(\mu_i,\sum_i))}{\partial \sum_i} = 0$ to get (9) and (10):

$$\mu^* = \frac{1}{N_i}\sum_{n=1}^{N_i} x^n \tag{9}$$

$$\sum^* = \frac{1}{N_i}\sum_{n=1}^{N_i}\left(x^n - \mu^*\right)\left(x^n - \mu^*\right)^T \tag{10}$$

So far, the distribution functions of each category have been determined. The predicted value can be obtained from the Bays formula as in (11):

$$\begin{aligned} P(C_1 \mid x) &= \frac{P(x|C_1)P(C_1)}{P(x|C_1)\,P(C_1) + P(x|C_2)\,P(C_2)} \\ &= \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}} \end{aligned} \tag{11}$$

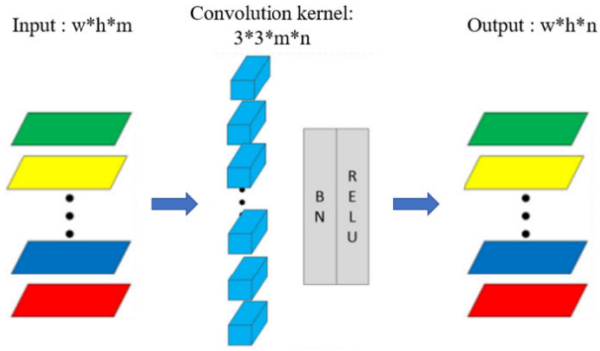



**FIGURE 3.** Traditional convolution neural network structure.

The prior probability distribution of each class can be brought into the Bayesian formula, as in (12):

$$P(C_1 \mid x) = \frac{1}{1+e^{-\left((\mu_1-\mu_2)^T\sum^{-1}x-\frac{1}{2}(\mu_1)^T\sum^{-1}\mu_1+\frac{1}{2}(\mu_2)^T\sum^{-1}\mu_2+\ln\frac{N_1}{N_2}\right)}} \tag{12}$$

Let $\omega = (\mu_1-\mu_2)^T\sum^{-1}$, $b = -\frac{1}{2}(\mu_1)^T\sum^{-1}\mu_1 + \frac{1}{2}(\mu_2)^T\sum^{-1}\mu_2 + \ln\frac{N1}{N2}$. Then the (12) is calculated and changed to (13):

$$P(C_1 \mid x) = \frac{1}{1+e^{-(wx+b)}} = \sigma(wx+b) \tag{13}$$

As shown in Fig. 3, the network adds a standardized processing to the input data of each layer during the training, named BN, in order to reduce the difference among samples. The activation function is named ReLu. It can overcome the problem of gradient disappearance and speed up the training speed.

### C. IMPROVEMENTS IN THE PROPOSED MOBILEVGG NETWORK

The traditional CNN model used by the VGG network is shown in Fig. 3, assuming that the input feature graph is in size and the dimension is $m$. In order to get a feature graph with output dimension $w^*h^*n$, the convolution kernel size is $3^*3^*m^*n$. Obviously, the number of parameters in need of realizing the traditional convolution is $9*m*n$. The more are kernels of the network, the more it occupies the memories.

The MobileVGG uses a three-layer bottleneck model which combines the detachable convolution and the point convolution to replace the conventional used by VGG16.

The three-layer bottleneck model is based on the Depthwise Separable Convolutions (DSC), which is a form of decomposition of convolution. It divides the traditional convolution into point convolution and depthwise convolution. The channels and ranges of the traditional convolution are changed in the DSC. It also can break the relation of the number of channels and the size of the kernels. It come true the separation of the channels and the ranges.

The point convolution is also named the pointwise convolution, of which kernel size is $1^*1$. The convolution kernel of the depthwise convolution is used in every channel. Then the $1^*1$ pointwise convolution combine all the output depthwise convolutions together. So, the DSC include the filter layer and the combine layer.

Standard convolutions can create new characteristic based on the filter and combine characteristics. The filter and combine operations will reduce the cost of the calculation by mean of the DSC. Meanwhile such decomposition process extremely decreases the calculated amount and the size of the model.

DSC consists of two parties, which are depthwise convolutions and pointwise convolutions. Depthwise convolutions are used to apply a convolution to the output channels. Then the pointwise convolutions are used to create the linear superposition of the depthwise layer output.

In every input channel of the depthwise convolutions. The kernel can be expressed as in (14):

$$G_{k,l,m} = \sum_{i,j} K_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \tag{14}$$

$K$ is the depthwise convolution kernel that the size is $D_k * D_k * M$.

As shown in Fig. 2, considering that replacing all convolution layers of VGG16 with three-layer structure will make the number of layers reach to 42. It will inevitably lead to gradient disappearance in the shallow middle network of the model and even make the network difficult to train and converge. Therefore, the MobileVGG uses the residual connection structure to solve this problem.

The full connection layer with a large number of parameters will be replaced by the $1^*1$ convolution. The maximum pooling dimension reduction in VGG16 is also removed. The dimension of the feature graph is adjusted by the step size of the convolution. Supposing the number of parameters required to implement the traditional convolution of $3^*3^*m^*n$ in VGG16 is $P_{VGG}$. The parameters required for the MobileVGG model is $P_{mVGG}$. The ratio of the two is $R$ as calculated in (15):

$$R = \frac{P_{VGG}}{P_{mVGG}} = \frac{9mn}{2mn+9n} \tag{15}$$

The $m$ represents the dimension of the convolution kernel. The convolution kernel used in this paper is $n = 4*m$, which is brought into the (15) to obtain (16):

$$R = \frac{36m^2}{8m^2+36m} = \frac{9m}{2m+9} \tag{16}$$

When a three-channel RGB image is entered, $m \geq 3$. So, $R > 1$, $P_{VGG} > P_{mVGG}$. Obviously, the MobileVGG network has fewer parameters in theory.

### D. THE NETWORK STRUCTURE OF THE PROPOSED MOBILEVGG

According to the description above, this section covers the details of the MobileVGG. As shown in Table 1,

**TABLE 1.** Network structure and parameters of the proposed MobileVGG.

| Layer | Input size | Convolution kernel dimension | Residual connection | Stride | Output size |
|---|---|---|---|---|---|
| **1** | 100*100*3 | 1*1*3*64,3*3*64,1*1*3*16 | No | 2 | 50*50*16 |
| **2** | 50*50*16 | 1*1*16*64,3*3*64,1*1*3*16 | Yes | 1 | 50*50*16 |
| **3** | 50*50*16 | 1*1*16*128,3*3*128,1*1*128*32 | No | 2 | 25*25*32 |
| **4** | 25*25*32 | 1*1*32*128,3*3*128,1*1*128*32 | Yes | 1 | 25*25*32 |
| **5** | 25*25*32 | 1*1*32*256,3*3*256,1*1*256*64 | No | 2 | 13*13*64 |
| **6** | 13*13*64 | 1*1*32*256,3*3*256,1*1*256*64 | Yes | 1 | 13*13*64 |
| **7** | 13*13*64 | 1*1*32*256,3*3*256,1*1*256*64 | Yes | 1 | 13*13*64 |
| **8** | 13*13*64 | 1*1*64*512,3*3*512,1*1*512*128 | No | 2 | 7*7*128 |
| **9** | 7*7*128 | 1*1*128*512,3*3*512,1*1*512*128 | Yes | 1 | 7*7*128 |
| **10** | 7*7*128 | 1*1*128*512,3*3*512,1*1*512*128 | Yes | 1 | 7*7*128 |
| **11** | 7*7*128 | 1*1*128*512,3*3*512,1*1*512*128 | No | 2 | 4*4*128 |
| **12** | 4*4*128 | 1*1*128*512,3*3*512,1*1*512*128 | Yes | 1 | 4*4*128 |
| **13** | 4*4*128 | 1*1*128*512,3*3*512,1*1*512*128 | Yes | 1 | 4*4*128 |
| **14** | 4*4*128 | 1*1*128*1024 | No | 1 | 4*4*1024 |
| **15** | 4*4*1024 | 1*1*1024*1024 | No | 1 | 4*4*1024 |
| \ | 4*4*1024 | Global average pooling | No | \ | 1*1*1024 |
| **16** | 1*1*1024 | 1*1*102483 | No | 1 | 1*1*3 |

the 13 convolution layers in VGG16 are replaced by the form of the point convolution, the depthwise convolution, and the point convolution. The form of each layer is shown in Fig.2. So, the total number of network layers reaches to 42. The main characteristic is compression of the model, which is easy to train and can lose as little feature information as possible.

The following are the details of the structure of the MobileVGG:

a) The combination of the point convolution, the depthwise convolution and the point convolution are used to reduce the number of parameters of the model.

b) The residual connection is used to overcome the problems of gradient disappearance and shallow training.

c) The convolution step size is adjusted to reduce the loss of information. The pool layer dimension is not reduced.

d) Instead of the full connection layer, the point convolution, and the global average pooling layer are used to reduce model parameters.

## V. EXPERIMENTS

Our method was implemented using Windows 10 + Tensorflow 1.2.0 + on a PC using an Intel Core i5-6500 3.20GHz processor and Quadro P4000, 8G RAM.

In the experiment, the training dataset uses the motion features based on the frame difference method to pre-detect the captured images. We collected our own datasets using HIKVISION DS2D3304IW-D4 Webcam, which was fixed 1.6meters above the ground. The self-collected dataset mainly contains three actions, including walking, falling on the ground, and sitting.

Among them, the training set includes 3628 pictures, which is mainly divided by the change of horizontal and vertical velocity of the human body. The numbers of the three behaviors are walking (1176), falling on chairs (1272) and falling to the ground (1180). The dataset is divided into the training dataset and the test dataset by a ratio of 1:1.

### A. THE COMPARATIVE EXPERIMENT OF ACCURACY BETWEEN THE MOBILEVGG AND VGG16

As shown in Fig. 4, there are each network trains and tests accuracy curves. Through the training and test accuracy of each model, we can find that the model can converge well. Fig. 4 (a) is based on the VGG16 pre-training model (the input size is 224*224), and Fig. 4 (b) is the VGG16 without the pre-training model (the input size is 100*100). Fig. 4 (c) is based on the MobileVGG_noRes without the residual connection (the input size is 100*100), and Fig. 4 (d) is MobileVGG with the residual connection (the input size is 100*100).

By comparing Fig. 4 (a) with Fig. 4 (b), it can be seen that loading the VGG16 pre-training model can make the model converge quickly. But it has a large number of parameters. Fig. 4 (c) and Fig. 4 (d) show that the convergence rate of the model is greatly affected by using the residual connection or not when iterating over the same Epoch.

The trends in Fig. 4 (c) and Fig. 4 (d) curves prove the necessity of using residual connections. The convergence speed and training test accuracy of networks without residual connections are worse than the networks with residual connections. From the comparison between Fig. 4 (d) and Fig. 4 (b), it can be seen that when the pre-training model is not loaded, the precision of MobileVGG can be closed to

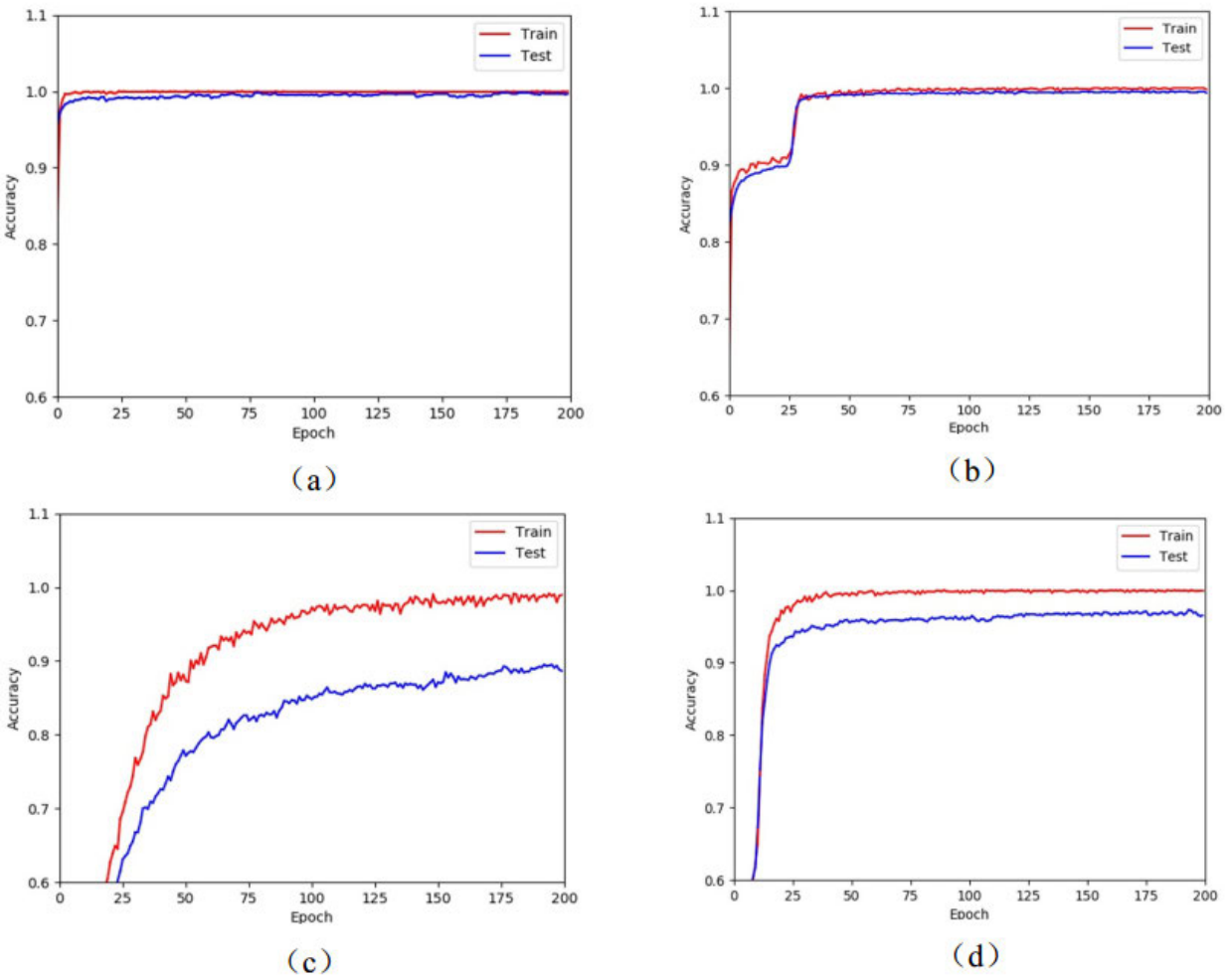



**FIGURE 4.** The training accuracy and test accuracy curves in each network training process. (a) Results when using VGG16 pre-training model. (b) Results when using VGG16 without pre-training model. (c) Results when using MobileVGG_noRes without residual connection. (d) Results when using MobileVGG with residual connection.

the precision of VGG16. The reduction of parameters just reduces few precisions of the MobileVGG.

### *B. COMPARISON OF CLASSIFICATION PERFORMANCE BETWEEN VGG16 AND THE MOBILEVGG*

As shown in Table 2, every method is compared in these aspects: the residual connection, training speed, test accuracy, and the memory occupied by the model. The hyperparameters of two groups during training are the same: Batch size=16, Epoch=200, Optimizer=Adam, and Learning rate=0.00001&0.000001. This learning rate indicates that the first 100 Epoch is 0.00001 and the last 100 Epoch is 0.000001. In each group of experiments, the picture input sizes are 224*224 and 100*100. The following is an analysis of the performance of each network model in terms of the memory occupied by the model, the speed of training and testing, test accuracy and the necessity of the residual connection.

In the memory occupied by the model, when the network hyperparameter Batch size is 16 and the input size is 224*224, the memory occupied by the VGG16 is 63 times as much as that occupied by the MobileVGG under the same condition. When the network hyperparameter Batch size is set as 16 and the input size is 100*100, the memory occupied by the VGG16 model is 28 times as large as that occupied by the MobileVGG under the same condition. As shown in Table 2, the identical property is performed when the Batch size is 32 and the input size is 100*100 or 224*224. The experimental results show that the improved lightweight MobileVGG has better performance in the memory occupied than the traditional VGG16 network. Less model memory makes the model no longer stay on the GPU-based desktop demonstration, making it possible to be used in the smart mobile phone.

In regard to the training and testing speed of the model, when the network hyperparameter Batch size is 16 and the input size is 224*224, the time of MobileVGG training, and testing are 14.05s and 3.06s. The time required for the training and testing based on VGG16 is 29.5s and 9.46s with the same conditions. When the input size change to 100*100, the time of MobileVGG training and testing are 3.69s and 0.79s while the VGG16 needs 6.53s and 2.92s. It can be

**TABLE 2.** Comparison of behavior classification performance of each network.

| Method | Input | Pre-training model | Residual connect | Training speed | Test accuracy | Test speed | Model memory |
|---|---|---|---|---|---|---|---|
| Hyperparameter: Batch size=16, Epoch=200, Optimizer=Adam, Learning rate=0.00001 & 0.000001 | | | | | | | |
| VGG16 | 224*224 | Yes | No | 29.46 | 0.9901 | 9.48 | 1555.7 |
| VGG16 | 100*100 | No | No | 6.53 | 0.9881 | 2.93 | 688.0 |
| MobileVGG_noRes | 224*224 | \ | No | 14.05 | 0.8907 | 3.06 | 24.7 |
| MobileVGG_noRes | 100*100 | \ | No | 3.69 | 0.9066 | 0.79 | 24.7 |
| Proposed MobileVGG | 224*224 | \ | Yes | **14.05** | **0.9793** | **3.06** | **24.7** |
| Proposed MobileVGG | 100*100 | \ | Yes | **3.69** | **0.9811** | **0.79** | **24.7** |
| Hyperparameter: Batch size=32, Epoch=200, Optimizer=Adam, Learning rate=0.00001&0.000001 | | | | | | | |
| VGG16 | 224*224 | Yes | No | 26.45 | 0.9913 | 8.69 | 1555.7 |
| VGG16 | 100*100 | No | No | 5.61 | 0.9883 | 2.26 | 688.0 |
| MobileVGG_noRes | 224*224 | \ | No | 13.33 | 0.8904 | 2.96 | 24.7 |
| MobileVGG_noRes | 100*100 | \ | No | 3.40 | 0.9145 | 0.75 | 24.7 |
| Proposed MobileVGG | 224*224 | \ | Yes | **13.33** | **0.9801** | **2.96** | **24.7** |
| Proposed MobileVGG | 100*100 | \ | Yes | **3.40** | **0.9825** | **0.75** | **24.7** |

found in Table 2 that their performances are the same when Batch size is 32 and the input size is 100*100 or 224*224. The experimental results reflect that the modified lightweight MobileVGG has improved the training and test speed compared with the traditional network VGG16.

In terms of the accuracy of the model, when the network hyperparameter Batch size is 32 and the input size is 224*224, the test accuracy based on the VGG16 pre-training model is 99.01%. And the test accuracy of the MobileVGG is 97.93% under the same conditions with a shortfall of 1.08%. When the Batch size is 16 and input size is 100*100, the test accuracy of VGG16 pre-training model is 98.81%, and the test accuracy of the MobileVGG is 98.11% having a shortfall of 0.70%. In Table 2, similar performance is shown when Batch size is 32 and the input size is 100*100 or 224*224. Although the test accuracy of the MobileVGG is slightly reduced, the VGG16 pre-training model needs fixed-size input. Such resize of the image will inevitably lead to the loss of image information. At the same time, the MobileVGG shows better performance in terms of the memory size and training and test speed.

In terms of whether the MobileVGG used residual connection structure, when the network hyperparameter Batch size is 32 and the input size is 224*224, the test accuracy of MobileVGG pre-training model based on the residual connection is 98.01%. The test accuracy of MobileVGG_noRes is just 89.04%, which is 8.97% lower than the former one under the same conditions. When the size is 32 and the input size is 100*100, the test accuracy of the MobileVGG pre-training model with residual connection is 98.25%. Other things being equal, the test accuracy of MobileVGG_noRes without the residual connection is 91.45%, which is 6.80% lower than the former one. As shown in Table 2, similar performance is shown when the Batch size is 16 and the input size is 100*100 or 224*224. The results show that the precision will be lower if not use the residual connection in the same iterations. It also will lead to the gradient disappeared and make gradient reflux blocked. Then it is hard to update the parameters and make the model difficult to train. To overcome above problems, just use the residual connection structure.

In summary, the proposed MobileVGG model demonstrates good performance in the memory occupied, training and testing speed, test accuracy, and so on. While maintaining the test accuracy, the model can reduce the memory compared traditional network and improve the speed of training and testing. Finally, the residual connection can restrain the gradient disappeared of the deep model and make the model converge better.

## VI. CONCLUSION

In this paper, a two-stream approach to fall detection with the MobileVGG is proposed. It first uses a stream based on motion characteristics to extract the video frames of possible fall behavior, and can effectively distinguish the fall from similar daily activities such as lying down. On the basis of the VGG network model, a lightweight network structure is put forward. The suspected fall video frames obtained by detection are sent to the MobileVGG model for classification. Finally, the accurate fall detection results are obtained. In the experimental results of the self-collected dataset, the input video frames can be accurately classified. The experimental results show that the proposed two-stream lightweight fall classification model outperforms the existing methods

in distinguishing the fall from similar daily activities like lying down and reducing the occupied memory. Hence, it is suitable for mobile devices.

In the future work, the proposed model will be further researched and improved so that the fall detection effect in the video can be further modified. The 3D network will be considered to be used in the model as well.

## 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*, vol. 6, pp. 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," *The Joint Commission J. Qual. Patient Saf.*, vol. 43, no. 9, pp. 460–470, Sep. 2017.

[3] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 290–301, Jan. 2015.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556* [Online]. Available: https://arxiv.org/abs/1409.1556

[5] R. Kamiya, T. Yamashita, M. Ambai, I. Sato, Y. Yamauchi, and H. Fujiyoshi, "Binary-decomposed dcnn for accelerating computation and compressing model without retraining," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 1095–1102.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[8] Y. Wang, K. Wu, and L. M. Ni, "WiFall: Device-free fall detection by wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 581–594, Feb. 2017.

[9] J. He, S. Bai, and X. Wang, "An unobtrusive fall detection and alerting system based on Kalman filter and Bayes network classifier," *Sensors*, vol. 17, no. 6, p. 1393, Jun. 2017.

[10] X. Kong, L. Meng, and H. Tomiyama, "Fall detection for elderly persons using a depth camera," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Xiamen, China, Dec. 2017, pp. 269–273.

[11] P. Tsinganos and A. Skodras, "A smartphone-based fall detection system for the elderly," in *Proc. 10th Int. Symp. Image Signal Process. Anal.*, Ljubljana, Slovenia, Sep. 2017, pp. 53–58.

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

[13] K. Ozcan, S. Velipasalar, and P. K. Varshney, "Autonomous fall detection with wearable cameras by using relative entropy distance measure," *IEEE Trans. Hum.-Mach. Syst.*, vol. 47, no. 1, pp. 31–39, Feb. 2017.

[14] A. Ejupi, M. Brodie, S. R. Lord, J. Annegarn, S. J. Redmond, and K. Delbaere, "Wavelet-based sit-to-stand detection and assessment of fall risk in older people using a wearable pendant device," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 7, pp. 1602–1607, Jul. 2017.

[15] B. Ando, S. Baglio, C. O. Lombardo, and V. Marletta, "An event polarized paradigm for ADL detection in AAL context," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 7, pp. 1814–1825, Jul. 2015.

[16] A. Mehmood, A. Nadeem, M. Ashraf, T. Alghamdi, and M. S. Siddiqui, "A novel fall detection algorithm for elderly using SHIMMER wearable sensors," *Health Technol.*, vol. 9, no. 4, pp. 631–646, Aug. 2019, doi: 10.1007/s12553-019-00298-4.

[17] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, and F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 1, pp. 349–361, Jan. 2020, doi: 10.1007/s12652-019-01214-4.

[18] W. Min, H. Cui, H. Rao, Z. Li, and L. Yao, "Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics," *IEEE Access*, vol. 6, pp. 9324–9335, 2018.

[19] S. S. Khan and J. Hoey, "Review of fall detection techniques: A data availability perspective," *Med. Eng. Phys.*, vol. 39, pp. 12–22, Jan. 2017, doi: 10.1016/j.medengphy.2016.10.014.

[20] W. Min, L. Yao, Z. Lin, and L. Liu, "Support vector machine approach to fall recognition based on simplified expression of human skeleton action and fast detection of start key frame using torso angle," *IET Comput. Vis.*, vol. 12, no. 8, pp. 1133–1140, Dec. 2018.

[21] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, "Plug-and-play CNN for crowd motion analysis: An application in abnormal event detection," in *Proc. 2018 IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Tahoe, CA, USA, Mar. 2018, pp. 1689–1698.

[22] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018, doi: 10.1109/access.2017.2778011.

[23] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5822–5831.

[24] S.-Y. Yan, Y.-D. An, J. S. Smith, and B.-L. Zhang, "Action detection in office scene based on deep convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICMLC)*, Jeju Island, South Korea, Jul. 2016, pp. 233–238.

[25] A. Nait Aicha, G. Englebienne, K. Van Schooten, M. Pijnappels, and B. Kröse, "Deep learning to predict falls in older adults based on daily-life trunk accelerometry," *Sensors*, vol. 18, no. 5, p. 1654, May 2018.

[26] M. D. Solbach and J. K. Tsotsos, "Vision-based fallen person detection for the elderly," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 1433–1442.

[27] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 314–323, Jan. 2019.

[28] Y. Wang, W. Zhou, Q. Zhang, and H. Li, "Enhanced action recognition with visual attribute-augmented 3D convolutional neural network," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, San Diego, CA, USA, Jul. 2018, pp. 1–4.

[29] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[30] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 314–323, Jan. 2019.

[31] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector CNNs," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2326–2339, May 2018.

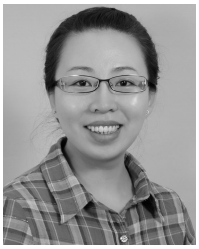
[32] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream CNN: Learning representations based on human-related regions for action recognition," *Pattern Recognit.*, vol. 79, pp. 32–43, Jul. 2018, doi: 10.1016/j.patcog.2018.01.020.

[33] G. Gkioxari and J. Malik, "Finding action tubes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Bostom, MA, USA, Jun. 2015, pp. 759–768.

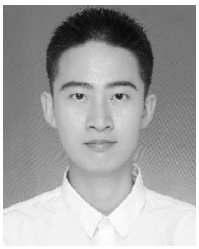
[34] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1933–1941.

[35] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Jun. 2014, pp. 568–576.

[36] W. Min, S. Zou, and J. Li, "Human fall detection using normalized shape aspect ratio," *Multimed Tools Appl*, vol. 78, no. 11, pp. 14331–14353, Jun. 2019.



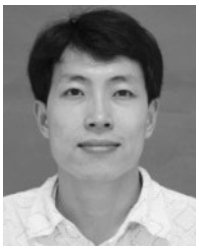
**QING HAN** received the B.E. and M.E. degrees in computer application from Tianjin Polytechnic University, China, in 1997 and 2006, respectively. She is currently an Associate Professor with the School of Information Engineering, Nanchang University, China. Her research interests include image and video processing and network management.



**HAOYU ZHAO** received the B.E. degree in computer science and technology from Nanchang University, China, in 2019, where he is currently pursuing the master's degree. His research interests include computer vision and deep learning.

**WEIDONG MIN** (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in computer application from Tsinghua University, China, in 1989, 1991, and 1995, respectively. He is currently a Professor and the Dean of the School of Software, Nanchang University, China. He is also an Executive Director of China Society of Image and Graphics. His current research interests include image and video processing, artificial intelligence, big data, distributed systems, and smart city information technology.

**HAO CUI** received the B.E. degree in computer science and technology from Nanchang University, China, in 2019, where he is currently pursuing the master's degree. His research interests include computer vision and deep learning.



**XIANG ZHOU** received the B.S. degree in computer science and technology and the M.S. degree in software engineering from Zhejiang University, China, in 2001 and 2005, respectively. He is currently a Lecturer with the School of Software, Nanchang University, China. His current research interests include computer vision and deep learning.

**KE ZUO** received the B.E. degree in animation and web design and the master's degree in art and design from Nanchang University, in 2006 and 2009, respectively. She is currently a Lecturer with the School of Software, Nanchang University. She mainly studies hybird design technologies, digital media art, and interaction design.

**RUIKANG LIU** received the B.E. degree in network engineering from the Jiangxi University of Science and Technology, China, in 2017. He is currently pursuing the M.E. degree with the School of Information Engineering, Nanchang University, China. His research interests include computer vision and deep learning.