IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 30, 2021, accepted September 16, 2021, date of publication September 17, 2021,
date of current version September 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113824

A Pose Estimation-Based Fall Detection Methodology Using Artificial Intelligence Edge Computing

WAN-JUNG CHANG[1], (Member, IEEE), CHIA-HAO HSU[1], AND LIANG-BI CHEN[2], (Senior Member, IEEE)

[1]Department of Electronic Engineering, Southern Taiwan University of Science and Technology, Tainan 71005, Taiwan
[2]Department of Computer Science and Information Engineering, National Penghu University of Science and Technology, Penghu 880011, Taiwan

Corresponding author: Liang-Bi Chen (liangbi.chen@gmail.com)

This work was supported in part by the Ministry of Science and Technology (MoST), Taiwan, under Grant MOST109-2221-E-162-001 and Grant MOST109-2622-E-218-009.

ABSTRACT As the population worldwide continues to age and the percentage of elderly people continues to increase, falls have been become the second leading cause of death from accidental or unintentional injuries. Although many imaging sensing devices have been used to detect falls for elderly people, most involve using the Internet to transfer images taken by a camera to a large back-end server, which performs the necessary calculations; however, algorithm limitations and computational complexity may cause bottlenecks in image outflow, and the image transfer can result in privacy problems. To address these problems, in this paper, an artificial intelligence (AI) fall detection method is proposed that operates using an edge computing architecture, called the pose estimation-based fall detection methodology (PEFDM), which is based on a human body posture recognition technique. The proposed PEFDM can effectively reduce the computational load, runs smoothly on mainstream edge computing systems and possesses artificial intelligence computing capabilities. By using edge computing, the privacy and upload bandwidth issues caused by image outflow can be eliminated. Experiments with real humans show that the PEFDM can detect falls for elderly people with a recognition accuracy of up to 98.1%.

INDEX TERMS Artificial intelligence over Internet of Things (AIoT), deep learning, edge computing, fall detection, Internet of Things (IoT), image recognition, image sensor application, posture recognition.

I. INTRODUCTION

Due to modern advances in medical treatments and public health, the average human life span has increased substantially. According to statistics from the World Health Organization (WHO), an American individual's average life expectancy is now 79.3 years and increases every year [1]. In this regard, the United States has created an aging society. As the average age of the population continues to increase, the number of accidents due to falls will also be increased [2]. Falls are now the second leading cause of elderly injuries and deaths. According to statistics from the U.S. Centers for Disease Control and Prevention [3], falls occurred for 25% of people over 65 years of age in the U.S. in 2012, causing 24,190 deaths and 3.2 million injuries and resulting in medical losses of US$26,340 and US$9,780, respectively [4].

According to the Kellogg international working group [5], the causes of falls can be classified as the result of an accident, a physical shock, stroke or epilepsy, consciousness disturbances, or a lower-level position. From a recognition perspective, according to references [6]–[8], a sudden decline in head height or a head remaining close to the ground for a substantial amount of time can be considered a fall.

In recent years, many devices have been created to detect and report falls in elderly individuals. These devices can be roughly divided into wearable systems and image recognition systems. Wearable devices are sometimes inconvenient to wear and need to be charged. Depending on the part worn and the conditions of use, the device could have an uneven accuracy [29]–[32]. Image recognition systems must transmit

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

images to a back-end computing platform due to the amount of calculations involved in fall recognition, causing privacy problems.

Therefore, to accurately detect and report falls in real time and address the privacy problems caused by image transmission, this paper proposes a fall detection method based on artificial intelligence edge computing; it combines image recognition with edge computing to avoid the limitations of local computing resources. In other words, we combine image recognition and edge computing to explore the use of local computing resources to replace cloud computing and solve privacy issues given the limited computing resources of edge computing devices. The efficiency of the algorithm can be adjusted to achieve specific goals. As a result, the proposed PEFDM provides the following main contributions:

- There is no need to use an expensive large computing server to infer the fall behavior, thus reducing the system construction cost.
- The computational complexity is greatly reduced so that the overall algorithm can run on edge computing devices with limited computing resources.
- Since only fall events are transmitted instead of video streams, the privacy issues caused by video outflow are greatly reduced.
- Same as above, it can also reduce the network bandwidth required by the overall system.

The remainder of this paper is organized as follows. Section II discusses related works on fall detection. Section III explains the proposed methodology. Section IV demonstrates and verifies the proposed methodology through experiments. Finally, Section V summarizes the results and discusses further research directions.

II. RELATED WORKS

Many works on methodologies for fall detection have been published [35]–[43]. For example, Mubashir *et al.* [33] noted that current fall detection technology can be divided into three categories: wearable device-based detection, environmental sensor-based detection and camera-based detection However, older people may not accept these related technologies into their lives due to an unfamiliarity with electronic devices and their associated privacy issues. Hence, overcoming these challenges is critical [34].

In this regard, Daher *et al.* [35] proposed a set of INRIA-Nanc sensing floors that can detect falls in elderly individuals. The INRIA-Nanc sensing floors consist of 104 intelligent tiles (60×60 cm^2), each of which uses a resistive force sensor to sense the walking, standing, sitting, lying, and ground-level falling behaviors of elderly individuals. If an individual is sitting on the ground, the system will use three-axis acceleration. The system is designed to comprehensively analyze whether any of these behaviors is a fall and then send the fall message, if generated, through the ZigBee wireless communication network.

Montanini *et al.* [36] proposed a set of wearable smart shoes for fall detection. The wearable intelligent shoes combine three resistive force sensors (FR1-3) with a three-axis acceleration sensor installed in the front and back of the insole, which are all connected to an embedded systems development board (Raspberry Pi 3) for algorithm calculation and Wi-Fi wireless communication transmission. Seventeen healthy subjects were subjected to a fall test in the laboratory, yielding an accuracy of 97.1%.

Hussain *et al.* [37] proposed a fall recognition system based on wearable sensors, combining an accelerometer and gyroscope to determine whether the wearer demonstrates falling behavior. Detection and recognition are divided into five stages: data acquisition, data preprocessing, feature extraction, fall detection, and fall activity recognition. The research combines traditional signal processing with machine learning classifiers (K-nearest neighbor, KNN) and detects falling behavior in elderly individuals. When an individual falls, the system analyzes the features of the individual before the fall, learns to recognize the fall activity, and establishes a model to predict the risk of future falls. Fall detection is performed with the KNN classifier, achieving a recognition accuracy of 99.8%.

Clemente *et al.* [38] proposed a real-time and nonintrusive indoor fall detection and real-time notification system. The system incorporates an intelligent vibration sensor in the environment, which uses the vibration generated when an individual falls on the ground, identifies the person who fell and immediately reports the location of the person. The proposed system uses two single classifiers (One-Class Support Vector Machine [SVM]) to classify features and analyze the vibration signals generated during walking. First, the first single classifier determines whether the signal corresponds to the one generated in the walking process. If not, the signal is sent to the second single classifier to determine whether the person has fallen.

Saadeh *et al.* [39] proposed a wearable fall detection and prediction system and applied it to patients. The system includes a three-axis acceleration sensor on the patient's thigh, which is used to distinguish the characteristics of the patient's activities of daily living from those of his or her falls. The proposed system is divided into two modes: 1).fall prediction, in which a nonlinear support vector machine classifier (NLSVM) is used to classify e seven features before the fall occurs to predict the fall before 300 ms $\sim$ 700 ms; and 2). fall detection, in which the Three-cascaded 1-sec sliding frames classification framework and offline training based on linear regression are used to establish different fall thresholds for different patients. According to the experimental results, the sensitivity of the system in fall prediction was 97.8%, the specificity was 99.1%, the sensitivity in fall detection was 98.6%, and the specificity was 99.3%.

Lee and Tseng [40] proposed a system that uses a built-in accelerometer to achieve fall detection. Specifically, the built-in three-axis acceleration characteristics of a smartphone are

used to detect the threshold values of three different attributes for the user while walking, running, and sitting. The system can detect four directions when falling, namely, forward and backward, left and right. When the user falls, the system immediately sends a warning message to the medical center for help. In an experiment, 650 different types of activities were tested, including 11 daily activities, with an accuracy of 99.38%.

Moulik *et al.* [41] proposed a fall detection system based on fuzzy theory, named FallSense, which uses an Arduino development board to connect a three-axis acceleration sensor with various sensors in the environment (infrared sensors, photodiodes) and an Ultrasonic sensor. The proposed FallSense system combines the advantages of the Internet of Things environment through the multiple data sensed by the different sensors and then uses fuzzy theory to infer whether the user has fallen in the environment. According to the experimental results, compared with a single sensor, this method of installing various sensors in the environment can reduce error reporting by 16% and can handle more complex situations in different environments.

Yu *et al.* [42] proposed a fall detection system based on the hidden Markov model (HMM). The authors used a three-axis acceleration sensor signal as the magnitude of acceleration in the direction of human movement as the criterion for judging a fall. In addition, they also integrated a direction calibration algorithm to reduce the judgment error caused by the inconsistent placement of the three-axis acceleration sensor. The system improves the difficulty encountered when using a HMM to detect falls (three-axis acceleration sensor, placement of the device and sensing direction) to improve the accuracy of fall recognition. The experimental results revealed a positive predictive value of 98.1% and a sensitivity of 99.2%, better than those reported in previous studies using HMM to detect falls.

With many studies using wearable devices to detect falls, Saleh and Jeannès [43] proposed a fall detection algorithm based on low-cost and high-precision machine learning that can effectively obtain the time when a fall occurs. The characteristic value of the algorithm addresses the misjudgment problem resulting from complicated environments during fall recognition. In this regard, the algorithm was developed using machine learning. The algorithm can be embedded in wearable devices to improve recognition accuracy and lower costs and power consumption. The experimental results revealed an accuracy exceeding 99.9% with calculations of less than 500 floating-point numbers per second, minimizing battery power consumption.

The above studies demonstrate that falls in elderly individuals are an increasingly severe public safety problem. However, compared with the present study, most fall detection was performed by installing sensors in the environment or allowing elderly individuals to wear devices on their bodies. These kinds of detection method are easily resisted by elderly individuals, however, because they are not accustomed to using electronic equipment [29]. This may be due to the inaccurate identification resulting from the complexity of the environment [34]. As a result, in this article, we will use images to judge the body.

Many academic studies have been published on fall recognition; however, due to hardware computing power limitations, most of them used simple methods to determine fall events. Lee and Mihailidis [9] proposed a fall detection system that used image processing technology. This method captures images via a suspended camera and sends them to a computer to perform background subtraction (BGS), which separates the background from pedestrians and then tracks the imaged object through connective component labeling (CCL). Finally, the perimeter of interest, Feret's diameter, and the image object's speed are judged based on posture, and the results are displayed onscreen. Twenty-one people of different ages (20–40 years) and heights (1.52 to 1.9 meters) were included to conduct a total of 315 experiments in a fixed environment and finally set two different thresholds based on height (threshold) for fall judgment. The authors calculated the true positives (TPs): 77%; false negatives (FNs): 23%; false positives (FPs): 5%; and true negatives (TNs): 95%. The total accuracy of this system reached 86%.

Cucchiara *et al.* [10] presented a fall detection and notification system based on HMM image processing technology. Multiangle images captured by multiple suspended cameras are transmitted to the computing system for BGS, shadow detection, image segmentation, and bounding box tracking to mark and track moving image objects. Then, the image is converted into waveform data through projection histograms, passed to the HMM for attitude recognition, and combined with data from multiple cameras to overcome situations in which the image object may be obscured from any single lens. Finally, a video streaming transcoding server (VSTS) is applied to transcode the video stream. The VSTS can be pushed to provide fall notifications to family members or caregivers.

In recent years, some related works [11]–[13] have combined advancements in imaging, computing methods, and hardware and applied machine learning (ML) as the basis for determining falls. Tran *et al.* [11] proposed a fall detection system that uses image processing and an SVM method to capture the room depth as well as an image using a Microsoft Kinect sensor (MKS) with an RGB-D camera and used the results to calculate the v-disparity [12].

Next, the Hough transformation is adopted to find edges with straight lines and obtain the location of the room floor in the image; then, deep learning is used to perform pose estimation (DLFPE). Twenty joint points from images of the human body are acquired through the MKS, and the body's distance from the ground, the vertical deviation angle, and the speed are calculated.

The study used the SVM to perform attitude recognition and obtain a final result. Six individuals of different ages (20–35 years old) were included to test nine action combinations (forward fall, backward fall, left fall, right fall, rapid sitting, lying down, walking, lifting objects, and sitting down)

twice each. A total of 108 consecutive images were subjected to fall identification verification. Finally, only the FP and TN were taken as references in a fixed environment; these values reached 3.3% and 27.7%, respectively.

Lie *et al.* [13] developed a fall recognition system that uses DLFPE and a long short-term memory (LSTM) deep learning model [14]. They used a suspended camera to capture pedestrian images and then performed calculations with DeeperCut [15] on fourteen recorded human body joint points (forehead, chin, both shoulders, elbows, wrists, hips, knees, ankles, and ears). The coordinates of these key nodes are input to the LSTM along with the coordinate array output by every set of eight images. The system recognizes five types of behavior: standing, walking, falling, lying down, and getting up.

Every eight consecutive images are divided into training data (800 units), verification data (255 units), and test data (250 units) for training and fall identification. Finally, the test data were used in a fixed experimental environment to perform five verification runs, achieving an average accuracy of 88.9%.

However, the calculations required for the abovementioned works are substantial; thus, they all send their images to a large, centralized computing device to perform these calculations. Unfortunately, this approach causes a number of problems, as described below.

A. SECURITY AND PRIVACY PROBLEMS

During image transmission, security and privacy issues can occur if the image passes through a router or switch to the back-end server. Such transmissions give criminals opportunities to obtain the surveillance camera images, which can result in serious security and privacy problems [44].

B. NETWORK BANDWIDTH PROBLEM

Network bandwidth problems can occur after the image is transmitted. Some algorithms require continuous, high-definition resolution images [11], [13]. When images are processed uniformly by a large computing device, the data volume may result in packet blockages and losses due to insufficient bandwidth. Ref. [45] used compressed images to address these privacy and bandwidth issues, but the images still needed to be decompressed server side to achieve normal image recognition.

C. COST PROBLEM

Large computing devices and system construction are expensive, difficult to replicate, and difficult to set up quickly, all of which affect the deployment of the associated fall detection devices, systems, and algorithms.

To address these problems, this article proposes an artificial intelligence (AI)-based fall detection method that operates on edge computing architecture like [55], namely, pose estimation-based fall detection methodology (PEFDM), which is based on recognizing human body postures. The proposed PEFDM runs smoothly on mainstream edge computing systems that possess AI computing capabilities.

III. THE PROPOSED PEFDM

A. HARDWARE MODULES: ADOPTION AND IMPLEMENTATION

1) IMAGE SENSING MODULE

Computer vision applications are widely used in automation. One indispensable module is the image module responsible for capturing images. Considering the shooting distances, angles, speeds, and image sizes required, we selected an image-sensing module (model: MIT-AA41PAF) made by MiTech as the camera for the AI fall image sensor and attached a 60 mm extended lens to adjust the image. The captured image's focal length and size make it similar to images captured by general surveillance cameras.

2) AI EDGE COMPUTING MODULE

Deep learning is a field that requires robust computing capabilities; a graphics processing unit (GPU) that can perform large-scale parallel computing is typically required. Computing acceleration software is needed on edge computing platforms to execute deep learning frameworks and deep neural networks. Therefore, the algorithm described in this study is based on software and hardware requirements evaluated and tested on two edge computing platforms, Jetson Xavier and Jetson TX2, both of which were developed by the NVIDIA Corporation. The Jetson Xavier is equipped with the latest Xavier system-on-a-chip (SoC); it uses an ARM architecture consisting of an 8-core processor with NVIDIA's Volta current-generation GPU architecture. The computing performance was the highest among the tested systems and is suitable for robotics applications and self-driving vehicles.

However, the Jetson Xavier requires 30 W of power, which is considerably more than the 15.5 W power supply available over Ethernet (PoE). This 15.5 W is the highest power used by general monitors, and such high power–consuming devices can cause heat dissipation problems. Finally, we selected the Jetson TX2 as the computing device in this study. The Jetson TX2 GPU uses the previous-generation Pascal architecture and is installed on a system with a CPU with a maximum of 6 cores, and a maximum of 8 GB of memory (RAM). A sample photograph and the specifications for the adopted image-sensing module and AI edge computing module are shown in Fig. 1.

Compared with traditional server-based image-sensing solutions, the proposed PEFDM is implemented with an AI edge computing module. Hence, the proposed PEFDM has a cost advantage over these traditional systems. Moreover, the proposed PEFDM is implemented or integrated as an Internet Protocol (IP) camera, which is also easy to use and install.

B. DESIGN OF THE PROPOSED HUMAN BODY POSTURE RECOGNITION-BASED PEFDM

The proposed PEFDM is divided into two main parts: 1) the human detection model (HDM), which is used to detect the position and skeleton of the human body in the image; and

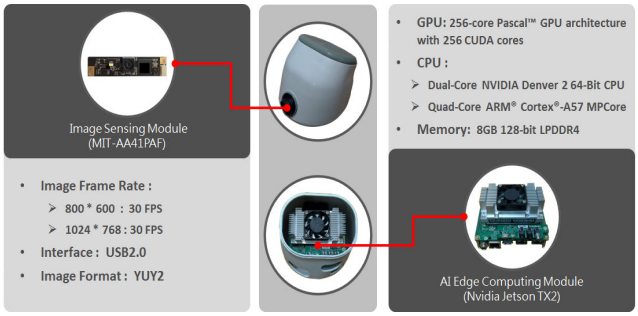

FIGURE 1. Photograph and specifications of the adopted image sensing module and AI edge computing module.

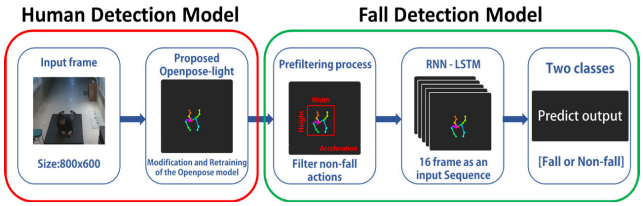

FIGURE 2. The process of the proposed PEFDM.

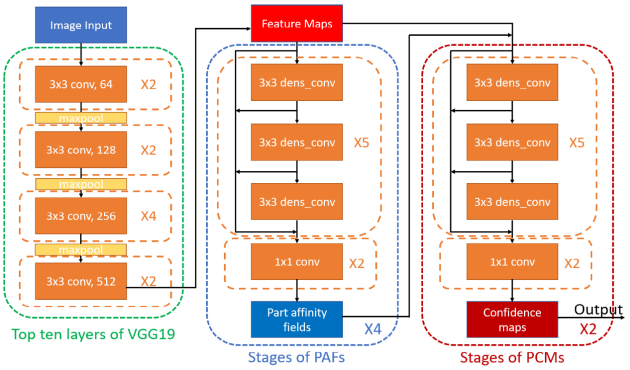

FIGURE 3. The OpenPose architecture.

FIGURE 4. The human body in the image produces blurred areas when moving fast, resulting in the loss of some joint points.

2) the fall detection model (FDM), which detects fall events based on positional changes of the human skeleton. The process of the proposed PEFDM is shown in Fig. 2. The two main modules in the proposed PEFDM are described below.

1) HUMAN DETECTION MODEL (HDM)

To recognize human behaviors, this work adopts deep learning pose estimation technology, which simultaneously detects an individual's skeleton and its position, making it beneficial for analyzing human movements. The goal is to implement this technology on an edge computing device with limited computing power. As the core technology for the proposed PEFDM, this work uses RAM and OpenPose [16], which have low computing requirements. The OpenPose architecture is shown in Fig. 3.

After confirming the performance of the OpenPose module based on the Visual Geometry Group-19 (VGG-19) model [17], which was run on a deep learning training server, we ported the module to the selected edge computing device (the NVIDIA Jetson TX2) to test and optimize its computing capabilities. The average number of frames per second (FPS) was 2.5; at this speed, the human body in the image produces blurred areas when moving fast, which can cause some joint points to be lost, as shown in Fig. 4.

To solve this speed problem while achieving the same or approximately the same accuracy, we referred to the module comparison benchmark proposed by Ref. [18]. Compared with the Lightweight-OpenPose proposed by Osokin [46], the feature extraction part of the VGG-19 model, which accounts for the largest computing time of the OpenPose module, was replaced with MobileNetV2, and other architecture modifications were performed to produce OpenPose-light. Unlike Lightweight-OpenPose, which uses the CPU to perform calculations and adjustments, our algorithm uses the GPU to perform the calculations. Therefore, the network reorganization method (refinement stages) used in Lightweight-OpenPose were not used in the current structure; instead, directly within the VGG-19 model, the CNN was replaced by the MobileNetV2 architecture, as shown in Fig. 5.

To calculate the speed difference in more detail, we use Multiply Adds and the neural network (NN) hierarchy proposed by MobileNet [19]. As shown in Table 1, we evaluated both of these modules, and the calculations and results are as follows.

- *MobileNetV2 [28]:* In Table 1, t is the expansion magnification of the inverted residual, c is the number of output channels, n is the number of repetitions, and s is the convolution offset stride for a total of 19 layers. Multiply Adds performs the following calculation (1):

$$\text{MultiAdds} = H * W * Cin * t * (Cin + k^2 + Cout), \tag{1}$$

  where H and W are the height and width of the input picture, respectively, C_{in} is the number of input channels, k^2 is the size of the convolution, and C_{out} is the number of output channels. The Multiply Adds total was 592,010,944.
- *VGG-19:* VGG-19 uses a general convolutional neural network (CNN), and its Multiply Adds calculation method is shown in (2).

$$\text{MultiAdds} = H * W * Cin * k^2 * Cout. \tag{2}$$

  The Multiply Adds total for VGG-19 is 11,271,536,640—approximately 19 times larger than that of MobileNetV2.

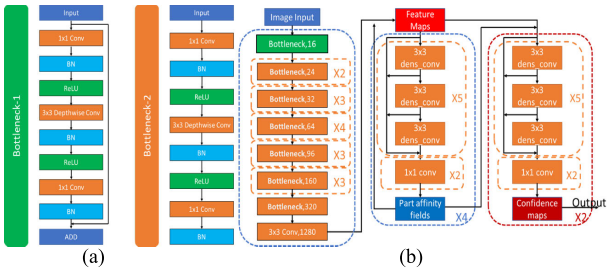

FIGURE 5. The architecture of the proposed OpenPose-light: (a) The bottleneck architecture of MobileNetV2; (b) The overall architecture of the proposed OpenPose-light.

TABLE 1. Network architecture of MobileNetV2 and VGG-19.

MobileNetV2						VGG19					
Input	Operator	t	c	n	s	Input	Operator	t	c	n	s
$224^2\times3$	Conv2d	_	32	1	2	$224^2\times3$	Conv2d	_	64	2	1
$122^2\times32$	bottleneck	1	16	1	1	$112^2\times64$	Conv2d	_	128	2	1
$122^2\times16$	bottleneck	6	24	2	2	$56^2\times128$	Conv2d	_	256	4	1
$56^2\times24$	bottleneck	6	32	3	2	$28^2\times256$	Conv2d	_	512	4	1
$28^2\times32$	bottleneck	6	64	4	2	$14^2\times512$	Conv2d	_	512	4	1
$14^2\times64$	bottleneck	6	96	3	1	$7^2\times512$	Conv2d1x1	_	4096	1	1
$14^2\times96$	bottleneck	6	160	3	2	$1^2\times4096$	FC	_	4096	2	1
$7^2\times160$	bottleneck	6	320	1	1	$1^2\times4096$	FC	_	1000	1	1
$7^2\times320$	Conv2d 1x1	_	1280	1	1	_	_	_	_	_	_
$7^2\times1280$	Avgpool 7x7	_	_	1	_	_	_	_	_	_	_
$1^2\times1280$	Conv2d 1x1	-	k	_	_	_	_	_	_	_	_

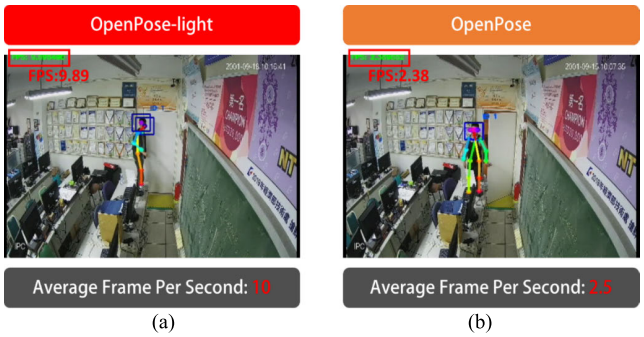

FIGURE 6. Module computing speed: (a) OpenPose-light; (b) OpenPose.

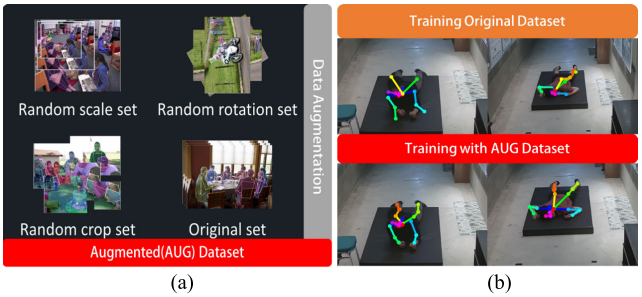

FIGURE 7. Module retraining: (a) Augmented (AUG) dataset; (b) Recognition results.

Feature extraction calculations account for approximately three-fifths of OpenPose's total calculations. Therefore, OpenPose-light can theoretically reach a speed of approximately 11 times greater than that of OpenPose. However, due to communication delays and bottlenecks between the CPU, GPU, and RAM, during real-world tests of the NVIDIA Jetson TX2, the true speed increased by only approximately 4 times, as shown in Fig. 6.

However, because OpenPose was trained on the CMU Panoptic Studio dataset [20], which lacks human fall motion samples, we used data enhancement methods (e.g., random scaling, flipping, and clipping) to compensate for the lack of fall data. The resulting recognition results are shown in Fig. 7. Table 2 shows a comparison of the mean average precision (mAP) [21] between the original dataset and the augmented dataset.

TABLE 2. Comparison of mAPs between original dataset and augmented dataset.

Dataset	mAP-50(%)	mAP-75(%)	mAP-85(%)
Origin dataset	81.5	62.6	58.4
Augmented dataset	**81.8**	**65.1**	**61.8**

TABLE 3. Comparison between the bending angle of the spine and the average number of triggering falls.

	FALLS	NONFALLS
5°	20	18.7
10°	20	12.2
15°	**20**	**4.1**
20°	17.3	3.5
25°	14.8	1.6
30°	10.4	1.1

2) FALL DETECTION MODEL (FDM)

The FDM consists of image preprocessing (prefiltering process) and an action recognition module, as shown in Fig. 8. The image preprocessing can mainly be divided into human body tracking, spine line deviation angle judgment, distance and acceleration judgment, and aspect ratio judgment. The purpose of the proposed FDM is to track and filter certain nonfalling actions to reduce the action recognition module's number of actions, reduce the number of calculations and costs and improve the smoothness of the image.

In human body tracking, we use a skeleton composed of 18 human body joint points output by OpenPose-light and track the human head with a centroid tracker [22]. This method calculates the center of gravity of the selected object and compares it with the next object. The speed of calculating the Euclidean distances to determine the center of gravity in the image is inversely proportional to the number of tracked objects. Because OpenPose-light predicts the human body's skeleton, some joint points may be lost due to ambient light or other factors. Therefore, compared with tracking the entire human body, tracking only the human head improves the stability and accuracy of the object's center of gravity.

A fall involves a continuous action of body tilting, acceleration and finally landing. Therefore, we referred to the spine line detection method proposed by Ref. [23] to determine the

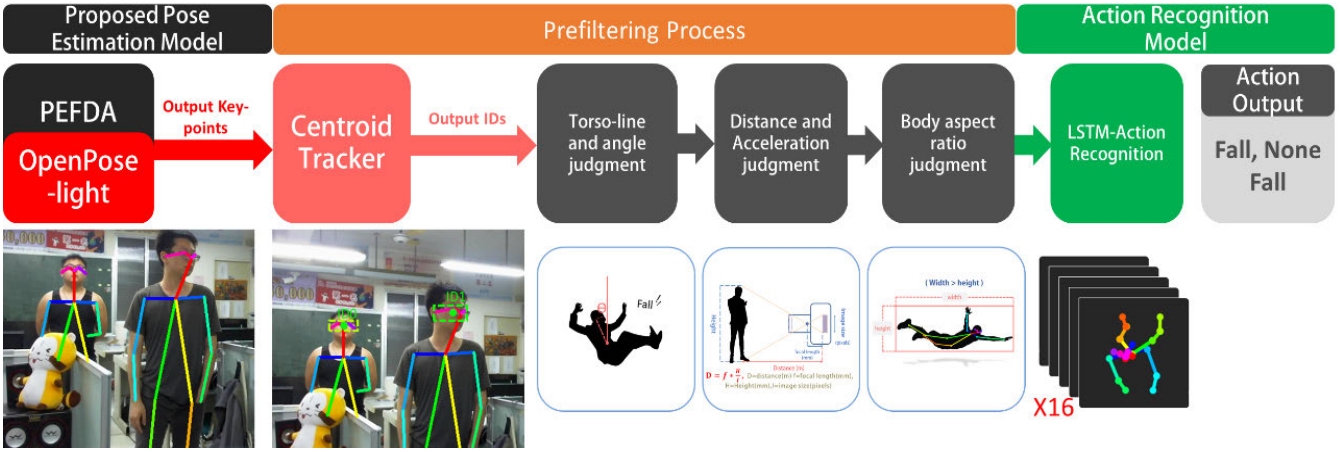

FIGURE 8. Process of the proposed FDM.

body tilt. Different from the horizontal lens and 3D coordinates used in Ref. [23], the lens device used in this work is located at a position similar to that of general surveillance cameras; thus, the image gravity line (IGL) captured by the camera lens may deviate from the real gravity line (RGL), and OpenPose-light outputs 2D coordinates, meaning that the depth of the spine line cannot be calculated.

Therefore, in this work, the spine line is calculated when the human body is detected for the first time. The algorithm stores the spine line ($\overline{FT}$) of the human body using the center points of the left and right hips and the left and right shoulders; then, it calculates the length according to the Euclidean distance formula, as shown in (3), and uses the law of cosines in (4) to calculate the offset angle T_a between this line and the human spine line ($\overline{ST}$) of the same human body, where $\overline{DT}$ is the distance between $\overline{FT}$ and $\overline{ST}$.

$$\overline{FT} = \sqrt[2]{\left(\frac{Hr_x + Hl_x}{2} - \frac{Sr_x + Sl_x}{2}\right)^2 + \left(\frac{Hr_y + Hl_y}{2} - \frac{Sr_y + Sl_y}{2}\right)^2} \tag{3}$$

$$T_a = arccos\frac{\overline{FT}^2 + \overline{ST}^2 - \overline{DT}^2}{2 * \overline{FT} * \overline{ST}} \tag{4}$$

When the deviation angle T_a is greater than a threshold value TH_a, the human body is determined to be falling forward. The decision process is shown in Fig. 8. After acquiring continuous images of 20 people falling, we determined that the algorithm is most sensitive and obtains the best results when TH_a is set to 15°, as shown in Table 3.

To perform distance and acceleration judgments, we referred to the image distance detection method proposed by Ref. [24], which calculates the distance from an object or target to the camera using the similarity between triangles. First, we determine the actual length or width of the object and calculate the actual distance according to the focal length of the lens (focal length) and the image size of the object, as shown in (5), where D is the true distance from the object to the lens (m), f is the focal length of the lens (mm), H is the actual length of the object (mm), and I is the length of the object in the image (pixels):

$$D = f * \frac{H}{I}. \tag{5}$$

After performing numerous measurements, we found that the key points of the head are not easily occluded and that the differences between individuals are small. The two ears and the nearest eye and ear are used as reference points for image ranging; these correspond to 1) the front view and rear view and 2) the oblique view and profile view, respectively.

After ranging, we can determine the distance between the human body and the lens. Then, we calculate the acceleration of the head when falling based on the center of gravity determined from the front and rear images and the acceleration formula shown in (6), where F_a is the acceleration of a falling head from the center of gravity, y_X is the y coordinate of the height of the center of gravity of the head in the front and rear images, and T is the sampling time (1/FPS).

Finally, using the universal gravitation formula, a human body's falling speed is the same as the G value of a free fall at 9.8 m/s. Nevertheless, after real-world testing, due to the deviation of the monitor's camera angle, in this work the falling threshold value A_{th} is set to 80% of the G value, or 7.84 m/s.

$$F_a = \frac{(y_2 - y_1) - (y_1 - y_0)}{t^2}. \tag{6}$$

The aspect ratio of the length of the human body to its width is larger when standing, walking, running, etc., except for certain special body shapes. In contrast, almost all lying-down movements cause the body aspect ratio to drop substantially. Therefore, in this study, the human body aspect ratio judgment proposed by Ref. [25] is used. By finding the maximum and minimum X- and Y-axis coordinates of 18 human body joint points, the object frame of the human skeleton (bounding box) is obtained, and its aspect ratio A_{sp} is calculated, as shown in (7). After the actual

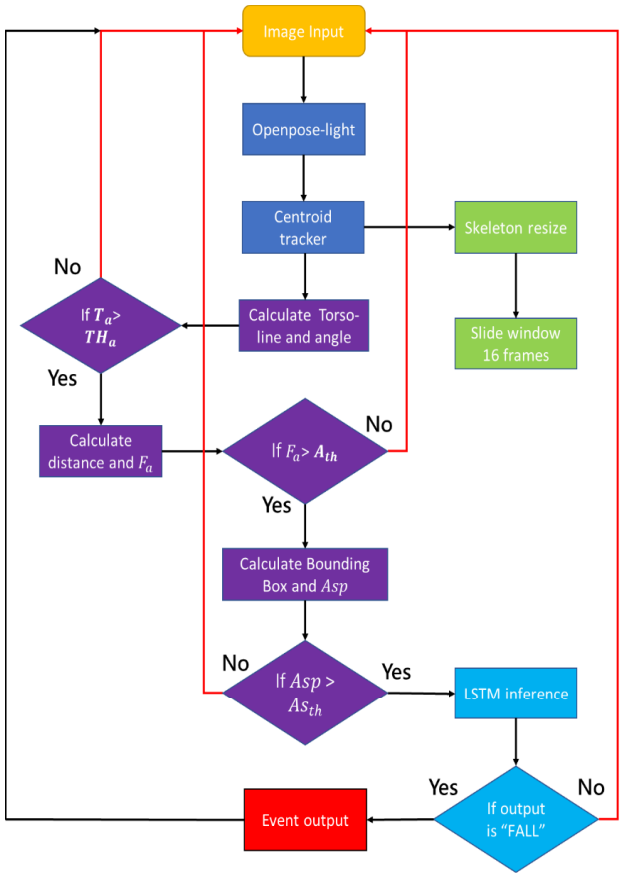

FIGURE 9. Flowchart of the proposed PEFDM.

fall test, because the purpose of this method is to reduce the number of calculations performed by the complete algorithm, too-stringent conditions will increase the error rate. For this study, we established a relatively loose threshold value for As_{th} by setting its value to $A_{sp} < As_{\text{th}} = 1.8$.

$$Asp = Height/Width. \tag{7}$$

Finally, the action recognition module uses an LSTM to perform fall recognition. We referred to the LSTM recognition posture method proposed by Ref. [13]. The human skeleton data are first presented separately and then resized, and a sliding window is used. The sliding window algorithm [26] stores the human skeleton in a first-in-first-out (FIFO) manner within an array of sliding windows. The FPS of the proposed PEFDM is 8, and the total time for a complete fall is approximately 2 seconds. Therefore, in this experiment, the LSTM judgment interval is set to 16 FPS, and the LSTM module established by Ref. [27] is for continuous image fall detection. A flowchart of the proposed PEFDM algorithm is shown in Fig. 9.

IV. EXPERIMENTAL DESIGN AND RESULTS

This section discusses the accuracy and computational efficiency of the proposed PEFDM. In the computational efficiency experiments, we executed OpenPose and OpenPose-light on the edge computing module NVIDIA Jetson TX2 and compared their computing efficiency scores to verify the effectiveness of the modified modules. For the accuracy experiments, we established an experimental environment, and the experimenters performed fall actions to verify the accuracy of the proposed PEFDM.

TABLE 4. Subject physical data.

	Subject	Height (cm)	Weight (kg)
Male	01	175	67
	02	172	63
	03	168	58
	04	174	45
	05	170	59
	06	167	97
	07	181	73
Female	08	179	72
	09	170	68
	10	165	45
Average		172.1	64.7

A. COMPUTATIONAL EFFICIENCY EXPERIMENTS

Our experimental environment executed OpenPose and OpenPose-light on an NVIDIA Jetson TX2 in a fixed environment to determine the calculation speed, memory consumption, and power consumption for each algorithm. We restarted the hardware after each test to ensure the same environment. As shown in Fig. 10, under the same power consumption, OpenPose-light consumed less memory and has a faster calculation speed than OpenPose. Therefore, the proposed PEFDM is more efficient.

B. FALL DETECTION ACCURACY EXPERIMENTS

First, we established both indoor and outdoor experimental environments. We installed the edge computing device for fall detection on a wall at a height between 2 and 2.5 meters, adjusted the viewing angle to focus on a distance of 4 to 12 meters, and placed a soft mattress in the fall test area to protect the subjects. Utilizing this fall test area environment, we then tested the OpenPose and OpenPose-light fall algorithms as follows.

Ten testers performed 4 different nonfall actions (walking, running, sitting, and squatting) and 4 fall actions (falling to the left, right, forward, and backward) 10 times, as shown in Table 4 and Fig. 11. Next, we recorded the recognition results of the two algorithms, including 1) the number of correctly recognized falls and 2) the number of correctly recognized nonfalls, and used (8) to calculate the accuracy (A_c) to measure the effectiveness of the algorithms.

$$A_c = \frac{Success}{Total} \times 100\% \tag{8}$$

Tables 5 and 6 show that neither algorithm produced false recognitions when predicting nonfalling actions; however, the proposed algorithm achieved excellent filtering effects. The LSTM module, which mainly performs fall

Model	Inference speed(FPS)	Memory usage(GB)	Power consumption(W)
Openpose-light	10.0	5.3	15
Openpose	2.5	6.6	15

FIGURE 10. Actual hardware performance test.

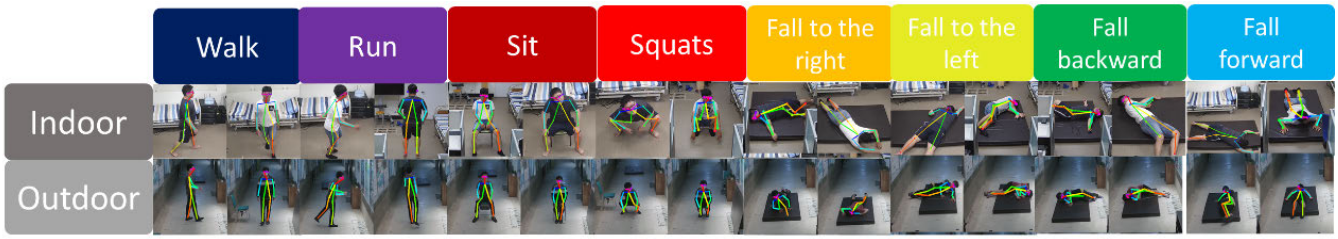

FIGURE 11. Actual fall test.

TABLE 5. Fall recognition test results based on OpenPose.

	Fall				No Fall			
Event / Count	Forward	Backward	Right	Left	Walk	Run	Sit	Squat
Success	81	67	77	82	100	100	100	100
Total	100	100	100	100	100	100	100	100
A_c	81%	67%	77%	82%	100%	100%	100%	100%
A_a	76.75%				100%			

TABLE 6. Fall recognition test results based on OpenPose-light.

	Fall				No Fall			
Event / Count	Forward	Backward	Right	Left	Walk	Run	Sit	Squat
Success	99	99	98	100	100	100	100	100
Total	100	100	100	100	100	100	100	100
A_c	99%	99%	98%	100%	100%	100%	100%	100%
A_a	99.0%				100%			

judgments, is different from other machine learning algorithms. The LSTM module can automatically learn features from sequence data, support multivariate data, and output variable-length sequences that can be used for multi-step prediction. Sexual movement inference has an excellent effect.

TABLE 7. Fall recognition test results by using three public datasets.

Accuracy (%)	OpenPose	OpenPose-light
Multiple Cameras Fall Dataset	85.7%	98.3%
UR fall detection dataset	83.2%	97.6%
Le2i fall detection dataset	84.6	98.1

For falls, we used OpenPose's fall recognition method, achieving an average rate of 76.75%, which is much lower than the 99% fall recognition rate using OpenPose-light. To further test the module, the Multiple Cameras Fall Dataset [47], UR fall detection dataset [48], and Le2i fall detection dataset [49] were used to test the fall recognition accuracy, and the results are shown in Table 7. The main reason for the superior results from OpenPose-light is that, as mentioned in Section III, the OpenPose calculation speed is too low, which causes the human body to produce fuzzy areas in the image when moving quickly and results in the loss of some joint points. These problems adversely affect the accuracy of fall recognition.

In addition, we also compared the results from other related studies that used human torso-based recognition

TABLE 8. Comparison the other related studies based on Le2i fall detection datasets.

Approach	Accuracy (%)	Execution Time(s)
Goudelis *et al.* [51]	96.6	-
Chamle *et al.* [52]	79.3	-
Poonsri *et al.* [53]	86.2	-
Alaoui *et al.* [54]	97.5	-
This Work	**98.1**	**0.1**

methods, as shown in Table 8. Compared with the results obtained by large-scale computing servers used in other related works. Ref. [50] adopts the history triple features technique reaches 96.6% accuracy; Ref. [51] uses the Adaboost classifier to achieve 79.3% accuracy; Ref. [52] adopts the Gaussian mixture model to achieve 86.2% accuracy. Ref. [53] uses Principal Component Analysis (PCA) combined with Support Vector Machine (SVM) reaches 97.5%. Compared [50]–[53], the precondition filter used in the proposed work with LSTM has higher accuracy. As a result, in this work, modules based on edge computing devices were used to obtain results of equal or greater accuracy.

V. CONCLUSION

This article addressed three major issues that occur in automated fall detection: security and privacy issues caused by transferring fall detection images over a network, network bandwidth issues caused by image transmission, and cost issues in system construction. We proposed a fall detection method based on artificial intelligence running on edge computing devices, which we called PEFDM. In contrast to previous fall detection approaches that require powerful computing devices, the detection method presented here is divided into two parts: an HDM, which uses OpenPose-light to recognize human joints, and an FDM, which uses human tracking, spine offset, distance and speed calculations to determine the aspect ratio used to eliminate continuous actions other than falls. This approach reduces the computational load on the edge computing device.

Finally, the human body image is processed through an LSTM, in which the continuous action of the skeleton junction points is used to make fall judgments. The proposed PEFDM algorithm is based on OpenPose-light. Consequently, we also compared the runtime performance of our fall recognition algorithm running on OpenPose-light on an NVIDIA Jetson TX2 with that of a native OpenPose fall algorithm, which has 4 times runtime speed and 25% more accuracy. As a result, experimental results has been demonstrated that the proposed PEFDM achieves up to 98.1% recognition accuracy. Compared to related works [50]–[53], the proposed PEFDM has higher accuracy.

Neither algorithm produced false recognition of any non-fall actions; however, the accuracy of OpenPose-light in detecting falling actions was much higher than that of OpenPose. The experimental results show that the proposed PEFDM can run smoothly on mainstream edge computing systems while exhibiting high accuracy and real-time judgment capabilities.

In the further work, considering the diversity and complexity of the environment, and to further strengthen the action recognition module based on LSTM, we will consider and evaluate to adopt reinforcement learning with long short-term memory (RL-LSTM)-based methodologies such as [56] and [57] to improve the proposed PEFDM. The human body key point data of known falls will be used to adjust and train the human body fall actions in an unknown environment.

REFERENCES

[1] World Health Organization (WHO). (2016). *World Health Statistics 2016: Monitoring Health for the SDGs Annex B: Tables of Health Statistics by Country, WHO Region and Globally*. [Online]. Available: https://www.who.int/gho/publications/world_health_statistics/2016/Annex_B/en/

[2] D. Chen, W. Feng, Y. Zhang, X. Li, and T. Wang, "A wearable wireless fall detection system with accelerators," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 2259–2263.

[3] National Council on Aging. (2018). *Falls Prevention Facts*. [Online]. Available: https://www.ncoa.org/news/resources-for-reporters/get-the-facts/falls-prevention-facts/

[4] E. R. Burns, J. A. Stevens, and R. Lee, "The direct costs of fatal and non-fatal falls among older adults—United States," *J. Saf. Res.*, vol. 58, pp. 99–103, Sep. 2016.

[5] R. O. Andres, L. C. Coppard, M. J. S. Gibson, and T. E. Kennedy, "Kellogg international work group on the prevention of falls by the elderly," Med. Faculties Univ. Copenhagen Aarhus, Danish Nat. Board Health Ugeskrift Læger Cooperation Univ. Michigan Univ. Copenhagen, Copenhagen, Denmark, Tech. Rep. PMID: 3595217, 1987. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/3595217/

[6] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3D head tracking to detect falls of elderly people," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Aug. 2006, pp. 6384–6387.

[7] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1274–1286, Aug. 2012.

[8] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *Proc. 10th Int. Conf. e-Health Netw., Appl. Services (HealthCom)*, Jul. 2008, pp. 42–47.

[9] T. Lee and A. Mihailidis, "An intelligent emergency response system: Preliminary development and testing of automated fall detection," *J. Telemed. Telecare*, vol. 11, no. 4, pp. 194–198, 2005.

[10] R. Cucchiara, A. Prati, and R. Vezzani, "A multi-camera vision system for fall detection and alarm generation," *Expert Syst.*, vol. 24, no. 5, pp. 334–345, Nov. 2007.

[11] T.-T.-H. Tran, T.-L. Le, and J. Morel, "An analysis on human fall detection using skeleton from Microsoft Kinect," in *Proc. IEEE 5th Int. Conf. Commun. Electron. (ICCE)*, Danang, Vietnam, Jul. 2014, pp. 484–489.

[12] C. Rougier, E. Auvinetn, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *Proc. Int. Conf. Smart Homes Health Telematics (ICOST)*, 2011, pp. 121–128.

[13] W.-N. Lie, A. T. Le, and G.-H. Lin, "Human fall-down event detection based on 2D skeletons and deep learning approach," in *Proc. Int. Workshop Adv. Image Technol. (IWAIT)*, Jan. 2018, pp. 1–4.

[14] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Jeju, South Korea, Feb. 2016, pp. 1–5.

[15] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "DeeperCut: A deeper, stronger, and faster multi-person pose estimation model," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 34–50.

[16] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[18] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, Oct. 2018.

[19] 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," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861
[20] (Apr. 18, 2021). *The CMU Panoptic Studio Dataset*. [Online]. Available: http://domedb.perception.cs.cmu.edu/
[21] Calculation of COCO key-point Dataset. (Apr. 18, 2021). *COCO mAP Evaluation*. [Online]. Available: https://cocodataset.org/#keypoints-eval
[22] J. C. Nascimento, A. J. Abrantes, and J. S. Marques, "An algorithm for centroid-based tracking of moving objects," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Phoenix, AZ, USA, Mar. 1999, pp. 3305–3308.
[23] L. Yao, W. Min, and K. Lu, "A new approach to fall detection based on the human torso motion model," *Appl. Sci.*, vol. 7, no. 10, p. 993, Sep. 2017.
[24] M. T. Bui, R. Doskocil, V. Krivanek, T. H. Ha, Y. T. Bergeon, and P. Kutilek, "Indirect method to estimate distance measurement based on single visual cameras," in *Proc. Int. Conf. Mil. Technol. (ICMT)*, May 2017, pp. 695–700.
[25] D. Bansal, A. Alsadoon, P. W. C. Prasad, M. Paul, and A. Elchouemi, "Elderly people fall detection system using skeleton tracking and recognition," *Amer. J. Appl. Sci.*, vol. 15, no. 9, pp. 423–431, Sep. 2018.
[26] F. Fumarola, A. Ciampi, A. Appice, and D. Malerba, "A sliding window algorithm for relational frequent patterns mining from data streams," in *Proc. 12th Int. Conf. Discovery Sci.*, 2009, pp. 385–392.
[27] G. Chevalier. (2016). *LSTMs for Human Activity Recognition*. [Online]. Available: https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition
[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[29] U. Lindemann, A. Hock, M. Stuber, W. Keck, and C. Becker, "Evaluation of a fall detector based on accelerometers: A pilot study," *Med. Biol. Eng. Comput.*, vol. 43, no. 5, pp. 548–551, Oct. 2005.
[30] T. Zhang, J. Wang, L. Xu, and P. Liu, "Using wearable sensor and NMF algorithm to realize ambulatory fall detection," in *Advances in Natural Computation*. Berlin, Germany: Springer, 2006, pp. 488–491. [Online]. Available: https://link.springer.com/chapter/10.1007/11881223_60#citeas
[31] R. Takeda, G. Lisco, T. Fujisawa, L. Gastaldi, H. Tohyama, and S. Tadano, "Drift removal for improving the accuracy of gait parameters using wearable sensor systems," *Sensors*, vol. 14, no. 12, pp. 23230–23247, 2014.
[32] H. Zhang, Y. Guo, and D. Zanotto, "Accurate ambulatory gait analysis in walking and running using machine learning models," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 191–202, Jan. 2020.
[33] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, no. 16, pp. 144–152, Jan. 2013.
[34] S. Kurniawan, "Older people and mobile phones: A multi-method investigation," *Int. J. Hum.-Comput. Stud.*, vol. 66, no. 12, pp. 889–901, Dec. 2008.
[35] M. Daher, A. Diab, M. El Badaoui El Najjar, M. A. Khalil, and F. Charpillet, "Elder tracking and fall detection system using smart tiles," *IEEE Sensors J.*, vol. 17, no. 2, pp. 469–479, Jan. 2017.
[36] L. Montanini, A. Del Campo, D. Perla, S. Spinsante, and E. Gambi, "A footwear-based methodology for fall detection," *IEEE Sensors J.*, vol. 18, no. 3, pp. 1233–1243, Feb. 2018.
[37] F. Hussain, F. Hussain, M. Ehatisham-Ul-Haq, and M. A. Azam, "Activity-aware fall detection and recognition based on wearable sensors," *IEEE Sensor J.*, vol. 19, no. 12, pp. 4528–4536, Jun. 2019.
[38] J. Clemente, F. Li, M. Valero, and W. Song, "Smart seismic sensing for indoor fall detection, location, and notification," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 524–532, Feb. 2020.
[39] W. Saadeh, S. A. Butt, and M. A. B. Altaf, "A patient-specific single sensor IoT-based wearable fall prediction and detection system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, pp. 995–1003, May 2019.
[40] J.-S. Lee and H.-H. Tseng, "Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8293–8302, Sep. 2019.
[41] S. Moulik and S. Majumdar, "FallSense: An automatic fall detection and alarm generation system in IoT-enabled environment," *IEEE Sensors J.*, vol. 19, no. 19, pp. 8452–8459, Oct. 2019.
[42] S. Yu, H. Chen, and R. A. Brown, "Hidden Markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 6, pp. 1847–1853, Nov. 2018.
[43] M. Saleh and R. L. B. Jeannès, "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 15, 2019.
[44] X. Zhang, S.-H. Seo, and C. Wang, "A lightweight encryption method for privacy protection in surveillance videos," *IEEE Access*, vol. 6, pp. 18074–18087, 2018.
[45] J. Liu, R. Tan, N. Sun, G. Han, and X. Li, "Fall detection under privacy protection using multi-layer compressed sensing," in *Proc. 3rd Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2020, pp. 247–251.
[46] D. Osokin, "Real-time 2D multi-person pose estimation on CPU: Lightweight OpenPose," in *Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, 2019, pp. 744–748.
[47] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple cameras fall dataset," DIRO-Université de Montréal, Montreal, QC, Canada, Tech. Rep. 1350, 2010, vol. 1350.
[48] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014. [Online]. Available: http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html
[49] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of arobust SVM based fall detection solution," in *Proc. 8th Int. Conf. Signal Image Technol. Internet Based Syst.*, Nov. 2012, pp. 218–224.
[50] G. Goudelis, G. Tsatiris, K. Karpouzis, and S. Kollias, "Fall detection using history triple features," in *Proc. 8th ACM Int. Conf. Pervas. Technol. Rel. Assistive Environ.*, Jul. 2015, pp. 1–7.
[51] M. Chamle, K. G. Gunale, and K. K. Warhade, "Automated unusual event detection in video surveillance," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Aug. 2016, pp. 1–4.
[52] A. Poonsri and W. Chiracharit, "Fall detection using Gaussian mixture model and principle component analysis," in *Proc. 9th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Oct. 2017, pp. 1–4.
[53] A. Y. Alaoui, S. El Fkihi, and R. O. H. Thami, "Fall detection for elderly people using the variation of key points of human skeleton," *IEEE Access*, vol. 7, pp. 154786–154795, 2019.
[54] W.-J. Chang, L.-B. Chen, M.-C. Chen, J.-P. Su, C.-Y. Sie, and C.-H. Yang, "Design and implementation of an intelligent assistive system for visually impaired people for aerial obstacle avoidance and fall detection," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10199–10210, Sep. 2020.
[55] R. Rajavel, S. K. Ravichandran, K. Harimoorthy, P. Nagappan, and K. R. Gobichettipalayam, "IoT-based smart healthcare video surveillance system using edge computing," *J. Ambient Intell. Hum. Comput.*, vol. 328, pp. 1–13, Mar. 2021, doi: 10.1007/s12652-021-03157-1.
[56] R. Rajavel and M. Thangarathanam, "Agent-based automated dynamic SLA negotiation framework in the cloud using the stochastic optimization approach," *Appl. Soft Comput.*, vol. 101, Mar. 2021, Art. no. 107040, doi: 10.1016/j.asoc.2020.107040.
[57] B. Bakker, "Reinforcement learning with long short-term memory," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synth.*, 2001, pp. 1475–1482.
[58] W. Song, C. Gao, Y. Zhao, and Y. Zhao, "A time series data filling method based on LSTM—Taking the stem moisture as an example," *Sensors*, vol. 20, pp. 1–21, Apr. 2020.

WAN-JUNG CHANG (Member, IEEE) received the B.S. degree in electronic engineering from Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2000, the M.S. degree in computer science and information engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2003, and the Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan, in 2008. He is currently a Professor with the Department of Electronic Engineering, Southern Taiwan University of Science and Technology. He is also the Director of the Artificial Intelligence over the Internet of Things Applied Research Center (AIoT Center) and the Internet of Things Laboratory (IoT Lab), Southern Taiwan University of Science and Technology. His research interests include cloud/IoT/AIoT systems and applications, protocols for heterogeneous networks, and WSN/high-speed network design and analysis. He received the Best Paper Award at IEEE ChinaCom 2009, the Best Paper Award at ICCPE 2016, the 1st Prize of the Excellent Demo! Award at IEEE GCCE 2016–2018, the Outstanding Paper Award from IEEE LifeTech 2020, and the First Prize Best Paper Award at IEEE ICASI 2017.

CHIA-HAO HSU received the B.S. and M.S. degrees in electronic engineering from Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in electronic engineering. His current research interests include deep learning and computer vision for the IoT and big data. He received the First Place Award from the Healthcare Applications Group at the 13th Digital Signal Processing Creative Design Contest, Ministry of Education (MoE), Taiwan.

LIANG-BI CHEN (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from the National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, in 2001 and 2003, respectively, and the Ph.D. degree in electronic engineering from Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2019. From 2004 to 2010, he was enrolled in the computer science and engineering Ph.D. Program at National Sun Yat-sen University, Kaohsiung. In 2008, he interned with the Department of Computer Science, National University of Singapore, Singapore. He was a Visiting Researcher with the Department of Computer Science, University of California at Irvine, Irvine, CA, USA, from 2008 to 2009, and the Department of Computer Science and Engineering, Waseda University, Tokyo, Japan, in 2010. In 2012, he joined BXB Electronics Company Ltd., Kaohsiung, as a Research and Development Engineer, where he was an Executive Assistant to the Vice President, from 2013 to 2016. In 2016, he joined Southern Taiwan University of Science and Technology, as an Assistant Research Fellow and an Adjunct Assistant Professor. In 2020, he joined the Department of Computer Science and Information Engineering, National Penghu University of Science and Technology, Penghu, Taiwan, as an Assistant Professor. He is a member of IEICE. He has served as a TPC member, an IPC member, and a reviewer for many IEEE/ACM international conferences and journals. He received the First Prize Best Paper Award from IEEE ICASI 2017, the Honorable Mention Award from IEEE ICAAI 2016, the First Prize Excellent Demo! Award from IEEE GCCE 2016–2018, and the Outstanding Paper Award from IEEE LifeTech 2020. He was a recipient of the 2014 IEEE Education Society Student Leadership Award, the 2019 Young Scholar Potential Elite Award of the National Central Library, Taiwan, and the 2018–2019 Publons Peer Review Award. He also received the 2020 Outstanding Associate Editor Award of IEEE Access Journal. Since 2019, he has been serving as an Associate Editor for IEEE Access and a Guest Editor for *Energies* and *Applied Sciences* journal.