System Based on Artificial Intelligence Edge Computing for Detecting Bedside Falls and Sleep Posture

Bor-Shyh Lin, *Senior Member, IEEE*, Chih-Wei Peng, I-Jung Lee, Hung-Kai Hsu, and Bor-Shing Lin, *Senior Member, IEEE*

***Abstract*—Bedside falls and pressure ulcers are crucial issues in geriatric care. Although many bedside monitoring systems have been proposed, they are limited by the computational complexity of their algorithms. Moreover, most of the data collected by the sensors of these systems must be transmitted to a back-end server for calculation. With an increase in the demand for the Internet of Things, problems such as higher cost of bandwidth and overload of server computing are faced when using the aforementioned systems. To reduce the server workload, certain computing tasks must be offloaded from cloud servers to edge computing platforms. In this study, a bedside monitoring system based on neuromorphic computing hardware was developed to detect bedside falls and sleeping posture. The artificial intelligence neural network executed on the back-end server was simplified and used on an edge computing platform. An integer 8-bit-precision neural network model was deployed on the edge computing platform to process the thermal image captured by the thermopile array sensing element to conduct sleep posture classification and bed position detection. The bounding box of the bed was then converted into the features for posture classification correction to correct the posture. In an experimental evaluation, the accuracy rate, inferencing speed, and power consumption of the developed system were 94.56%, 5.28 frames per second, and 1.5 W, respectively. All the calculations of the developed system are conducted on an edge computing platform, and the developed system only transmits fall events to the back-end server through Wi-Fi and protects user privacy.**

***Index Terms*—Bedside fall, deep learning, edge computing, neuromorphic computing hardware, sleep posture recognition.**

I. Introduction

According to the United Nations, the elderly population is projected to triple by 2085 [1]. Because of insufficient nursing staff and resources, elderly patients might face risks such as bedside falls when getting out of bed without assistance. Statistics indicate that most indoor fall events (∼66%) occur in the bedroom, with approximately 80.1% of these events occurring around the bed [2]. Such falls may even cause death [3]. Further, bedridden older adults face a high risk of pressure ulcers, which are a serious disease caused by remaining in the same posture for a long period [4]. Thus, the development of a bedside monitoring system that can function all day has become essential. Studies have placed wearable devices with inertial measurement units (IMUs) on the human body to measure human movements [5], [6], [7]. Jähne-Raden et al. proposed a system that includes a wearable device with an IMU that is worn on the patient's thigh [6]. This sensor detects leg movements indicative of a fall event [6]. The method of using a threshold value is simple and effective; however, users might feel uncomfortable wearing the aforementioned device or forget to wear the device. Some studies have used ambient devices, such as pressure sensors [8], [9], [10] or radio frequency (RF) devices [11], [12], [13], to monitor in-bed postures or bedside events. Matar et al. adopted the oriented gradients histogram and an algorithm based on local binary pattern to extract features from pressure maps. They then input these features into a feedforward neural network for posture classification [8] and achieved a classification accuracy of 97.9%. However, the relatively high cost of pressure sensing arrays hinders the use of the aforementioned method and renders it unsuitable for mass adoption. RF sensing devices provide a

Manuscript received 6 October 2022; revised 29 January 2023; accepted 24 April 2023. Date of publication 28 April 2023; date of current version 3 July 2023. This work was supported in part by the Ministry of Science and Technology in Taiwan under Grants MOST 109-2221-E-305-001-MY2, MOST 110-2314-B-305-001, and MOST 110-2221-E-A49-096-MY3, in part by the University System of Taipei Joint Research Program under Grants USTP-NTPU-NTOU-111-01 and USTP-NTPU-NTOU-112-01, in part by Faculty Group Research Funding Sponsorship by National Taipei University under Grant 2022-NTPU-ORDA-03, and in part by National Taipei University, Taiwan through Academic Top-Notch and Features Field Project under Grant 111-NTPU_ORDA-F-004. *(Corresponding author: Bor-Shing Lin.)*

Bor-Shyh Lin is with the Institute of Imaging and Biomedical Photonics, National Yang Ming Chiao Tung University, Tainan 71150, Taiwan (e-mail: borshyhlin@gmail.com).

Chih-Wei Peng is with the School of Biomedical Engineering, College of Biomedical Engineering, Taipei Medical University, Taipei 11031, Taiwan, and also with the School of Gerontology and Long-term Care, College of Nursing, Taipei Medical University, Taipei 11031, Taiwan (e-mail: cwpeng@tmu.edu.tw).

I-Jung Lee is with the Department of Computer Science and Information Engineering, National Taipei University, New Taipei City 237303, Taiwan, and also with the College of Electrical Engineering and Computer Science, National Taipei University, New Taipei City 237303, Taiwan (e-mail: ijunglee0215@gmail.com).

Hung-Kai Hsu and Bor-Shing Lin are with the Department of Computer Science and Information Engineering, National Taipei University, New Taipei City 237303, Taiwan (e-mail: alan840706@gmail.com; bslin@mail.ntpu.edu.tw).

Digital Object Identifier 10.1109/JBHI.2023.3271463

noninvasive method for monitoring human motion [11], [12], [13]. However, their detection performance depends largely on their location and orientation as well as occupant movement; thus, the aforementioned devices do not have high accuracy. Some vision-based bedside monitoring systems have been proposed in previous studies [14], [15], [16]. Chung et al. proposed a bedside monitoring method based on a single-chip system with an RGB camera [14]. When using an RGB camera, low power consumption is achieved; however, the system performance is considerably affected by the light condition. In [15], an RGB camera was replaced with a night vision camera for human monitoring; however, the high-resolution grayscale images generated by night vision cameras cause privacy concerns. If the RGB camera of a human motion monitoring system is replaced by a depth camera, privacy can be protected [16]; however, a depth camera sensor must continuously emit infrared signals and thus has high power consumption, which limits its application. Thermopile array sensing elements have attracted attention in bedside monitoring [17], [18], [19], [20]. Because thermal images do not appear a user's facial features, user privacy can be maintained when using these images for fall detection. At the same time, the thermopile array sensing element has the advantages of low power consumption and not easily affected by light compared with the depth camera. Therefore, a bedside monitoring system with thermopile array sensing elements was developed in this study. Conventional bedside monitoring systems use adequate and powerful back-end equipment to analyze the data collected by them [7], [8], [9], [10], [11], [12], [13], [15], [16], [17], [18], [19]. However, those devices required to upload large amount of data to the back-end server, causing higher costs on the larger network bandwidth. Transmission of large amount of data between the devices and the back-end server also caused higher latency regarding the transmission time. Moreover, the privacy concerns of uploading the images to the back-end server is still skeptical [21]. Therefore, edge-computing-based bedside monitoring systems have been developed [5], [6], [14], [20]. In these systems, the computations that must be performed are transferred from the back end to the front end. However, the limited computational resources and memory at the front end have resulted in the use of less robust algorithms, such as statistical methods and thresholds, for completing the computations [5], [6]. Some edge-computing-based studies have used image computing methods to extract human features [14], [20]. In these studies, the difference between images was used to determine the image foreground; thus, the methods used in the aforementioned studies are sensitive to background changes and are time-consuming. Asbjørn et al. developed a system for collecting data by using thermopile array sensing elements and ultrasonic sensors simultaneously to achieve high accuracy [20]. However, this system only detects out-of-bed events, and its execution time of 10 s might hinder emergency handling in the event of a fall. Studies have indicated that artificial intelligence (AI) edge computing can effectively replace back-end computing in fall detection systems and has high robustness [21], [22].

Considering that the fall detection system is highly relevant to this study, the method proposed in this study is based on the research proposed by Lin et al. [22]. Compared with pedestrian fall detection, bedside monitoring system faces more practical challenges. In addition to human motion, the interaction between the people and the bed is also considered, and fall events and posture classification are simultaneously monitored.

In this study, we attempted to porting the algorithm to a platform based on edge computing, which can quickly detect the bedside fall event to save time for helping the faller in advance and accurately detects the in-bed postures to prevent bedsores problem. Since the captured images are not transmitted to the back-end computer, the bedside monitoring system does not require additional bandwidth, which can provide both privacy protection and cost saving. The contributions of this study are listed as follows. First, the interference time of this study is lower than the other related studies because it does not require to transmit the data between the sensor side and the PC. Second, the system proposed in this study does not require high throughput internet connection to transmit the images because it only sends the fall events back to the central control terminal. Third, the power consumption of the core in this study is lower than the other studies, which is only 1.5 W.

II. Methods

A. System Overview

The entire operation of the developed system is illustrated in Fig. 1. The edge computing platform continually captures thermal images through the thermopile array sensing element. After collecting the thermal image, the neural network model deployed on the AI chip detects the first-stage posture, the bounding box of the human, and the bounding box of the bed in the image. Next, the CPU of the AI chip uses the vertical distances from the center of the bounding box of the human to the upper and lower edges of the bed to correct the first-stage posture to obtain the final-stage posture [17]. A fall event is defined as the state of the subject's body changing from the non-falling to the fall state and staying in the falling state for a period of time. When the fall event is detected, the system immediately transmits the fall event to the central control terminal through Wi-Fi for warning.

B. Hardware Description

Fig. 2 shows the hardware using for the bedside fall detection platform. The hardware for the platform is composed of a thermopile array sensing element (Lepton 3.5, Teledyne Technologies, Wilsonville, OR, USA) and an AI development board (Sipeed Maix GO, Sipeed Technology, Shenzhen, China). The Lepton 3.5 is a long-wave infrared camera module with a resolution of 160 × 120 pixels, a horizontal field of view of 56°, and a diagonal field of view of 71°. This sensor captures infrared radiation input from the nominal response wavelength band (8–14 μm) and converts it into electrical signals.

When using a thermopile array sensing element to capture a sleeping human body, the bed surface in contact with the human body has a higher temperature than the surrounding bed surface, which affects the human body contour in the thermal image. Therefore, the human body contour should be separated from the original image for posture classification with object detection algorithm. Many lightweight object detection algorithms have

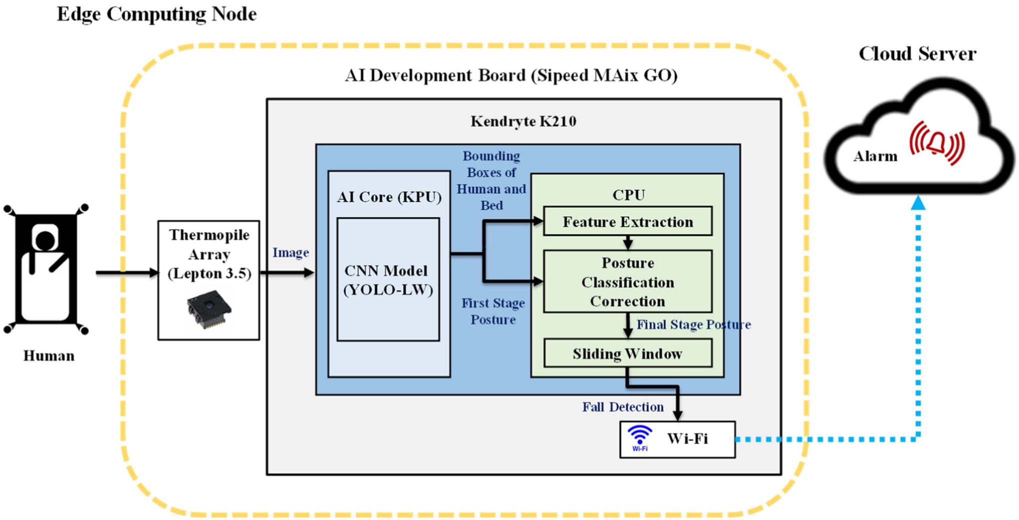

Fig. 1. Overall system architecture.

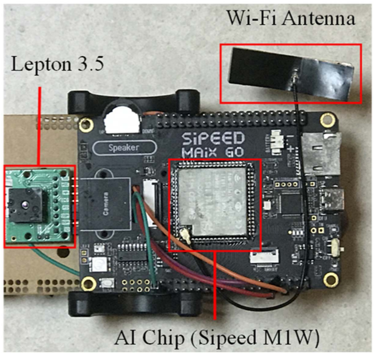

Fig. 2. Hardware of the proposed edge computing platform.

been developed [22], [23], [24], including robust AI models with high noise immunity that can extract the human body from an image background. To increase the speed of inferring AI models with an edge computing platform, some related studies have utilized graphics processing units (GPUs) [25] or field programmable gate arrays (FPGA) [26] with AI chips because the operation of AI model is similar to the operation of image processing. Considering that a bedside monitoring system must operate for a long duration, a high-power-consumption GPU is not suitable for the application because it increases the system cost. A FPGA might be the alternative because of its low power consumption; however, the cost of the hardware is too high to be adopted for the bedside monitoring application. Therefore, this study adopted the Sipeed M1W core module of the AI development board. The AI development board consists of an AI chip (Kendryte K210, Canaan Inc., Beijing, China) and a Wi-Fi chip (ESP8285, Espressif Systems, Shanghai, China). The power consumption of the AI chip is only 0.3 W, and it can perform 0.25 and 0.5 tera operations per second at frequencies of 400 and 800 MHz, respectively. In the present study, to reduce the training time, the aforementioned neural network was trained on a personal computer (PC) with a GPU (RTX 3080 10G, AsusTeK, Taipei, Taiwan), and the best AI model was determined through several experiments. All the tested models in this study are built on TensorFlow framework [27]. After determining the best AI model, the best AI model was quantized because AI chips only partially support operators in TensorFlow. The best AI model with supported operators is then deployed on the proposed bedside monitoring platform.

C. AI Algorithm for Posture Classification on the Edge AI Board

The process of the posture classification is presented in Fig. 3. After obtaining the thermal images, the first-stage posture, the position and bounding boxes of the subject and the bed, is extracted by the AI model. The posture of the subject is classified into one of the 12 postures, including standing, sitting, falling, and nine sleeping postures. After that, the features based on the relation of position of the subject and the bed are extracted for correcting the result of posture classification.

To implement on the AI development board, the lightweight object detection algorithms from previous studies [22], [23], [24] were adopted. You Only Look Once (YOLO) v3 or YOLO v3-tiny [28] is adopted as the network of backbone for extracting the features. The architectures of YOLO v3-tiny and YOLO-LW are displayed in Figs. 4 and 5, respectively. The YOLO v3-tiny model was first proposed in 2018. This model is an improved version of Darknet-53 [29] and can conduct multiscale prediction. The YOLO-LW algorithm proposed by Lin et al. is an improved version of YOLO v3-tiny [22] and detects the human body accurately even in a low-precision format. The depth-wise separable convolutional layers and simplified neurons [30] in YOLO-LW reduced the model parameters used by edge computing platforms with limited memory. The YOLO-LW algorithm has been successfully implemented on the edge computing platform of the Kendryte K210 AI chip. This model's input layer and output layer were modified to conduct object detection and sleep posture classification. To transfer the tasks originally performed in the back end to the edge computing platform, a small and efficient network model must be constructed for the constrained

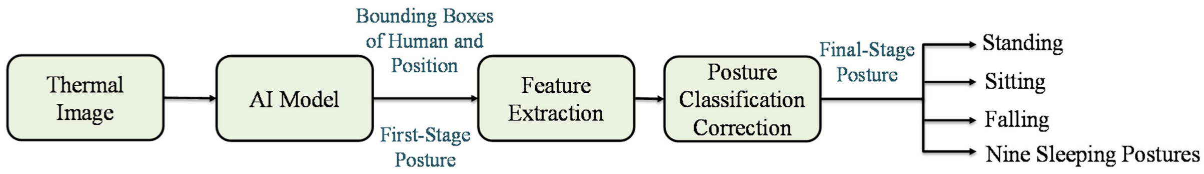

Fig. 3. Classification process performed by the software in the AI board.

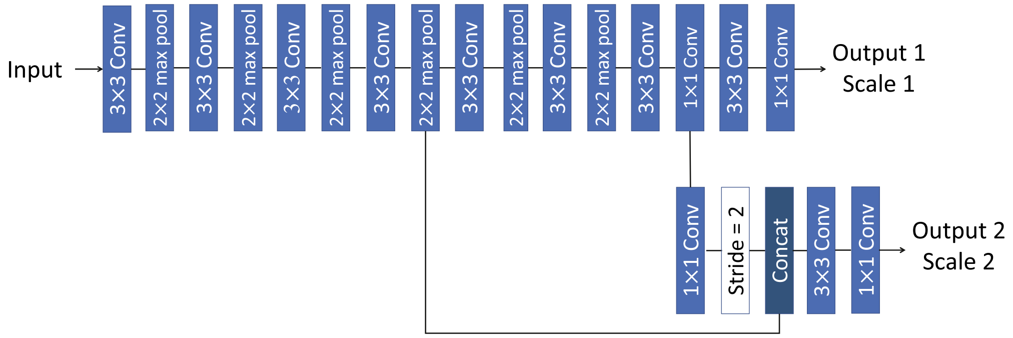

Fig. 4. Network structure of YOLO v3-tiny.

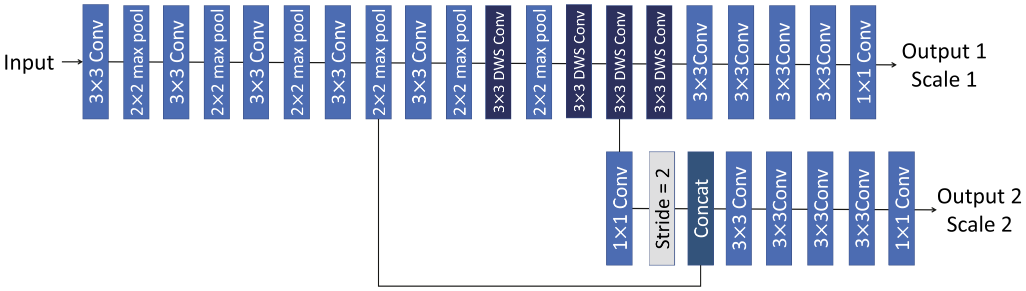

Fig. 5. Network structure of YOLO-LW.

environment. Methods for model compression and acceleration mainly include network pruning [31], network quantization [32], and knowledge distillation [33]. Among the aforementioned methods, network quantization enables the highest reduction in the number of network parameters. In network quantization, the model size is reduced using a relatively low-precision format to represent neuron parameters [34]. In the present study, the size of the YOLO-LW model was reduced to one-fourth its original size through quantization. For each neuron, the weights in the 32-bit floating-point format ("float 32") were converted into the int 8 format by using (1)–(4), and the bias was discarded.

$$\omega_q = f_r\left(Z + \frac{\omega}{S}\right) \tag{1}$$

$$Z = f_r\left(-\frac{\omega_{\min}}{S}\right) \tag{2}$$

$$S = \frac{\omega_{\max} - \omega_{\min}}{2^{bit\_width} - 1} \tag{3}$$

$$f_r(x) = \begin{cases} 0, & x < 0 \\ round(x), & 0 \le round(x) \le 2^{bit\_width} - 1 \\ 2^{bit_{width}} - 1, & x > 2^{bit\_width} - 1 \end{cases} \tag{4}$$

where ω is an element of the weight vector, ω_q is the quantized value,bit_width is the number of bits after weight format conversion, $\omega_{\max}$ is the highest-value element of the weight vector, and $\omega_{\min}$ is the lowest-value element of the weight vector. In (1), the original weight is scaled using the scale factor S and zero point Z, which is linearly mapped to the range of the int 8 format. Because YOLO-LW uses the leaky rectified linear unit as the activation function, the activation value is likely to have an uneven distribution; therefore, linear mapping cannot be used. According to the distribution of activation values in the test set, selective discarding activation values, which occur less frequently but seriously affect activation value distribution after conversion. The activation value of each neuron is calculated using (5)–(7).

$$t_q = f_r\left(Z + \frac{t}{S}\right) \tag{5}$$

$$Z = f_r\left(\frac{U}{S}\right) \tag{6}$$

$$S = \frac{2U}{2^{bit\_width} - 1} \tag{7}$$

where t is the activation value, t_q is the transformed activation value, and U is the transition threshold. To find a suitable transition threshold U, YOLO-LW is used to infer the test set, the activation value generated during the inference process of each neuron is recorded, and t_a and M are calculated using (8) and (9), respectively.

$$t_a = |t| \tag{8}$$

$$M = Max(|t_{\max}|, |t_{\min}|) \tag{9}$$

where t is the activation value, $t_{\max}$ is the maximum activation value produced by a neuron, and $t_{\min}$ is the minimum activation value produced by the neuron.

When dividing 2048 blocks between 0 and M , all t_a values of a neuron are assigned to corresponding blocks, and the medians from the 128th block to the 2048th block are sequentially selected as candidate thresholds (U_c). Next, the probability distribution P of all t_a values from the first block to the U_c value is calculated, and all t_a values are then converted into the int 8 format by using (4). Subsequently, the probability distribution Q of the converted t_a values from the first block to the U_c value is determined. Moreover, Kullback–Leibler divergence is used to estimate the difference between P and Q (10). Finally, the value of U_c corresponding to the smallest Kullback–Leibler divergence is selected as the final transition threshold U.

$$D\ (X, p, q) = -\sum_{x \in X} p\,(x)\,(log_2 q\,(x) - log_2 p\,(x)) \tag{10}$$

where X is a set of all blocks, x is a block, $p(x)$ is the probability of x on P, and $q(x)$ is the probability of x on Q.

The scale factor S and zero point Z of each neuron are used as prior knowledge to increase the inferencing speed of the int 8 neural network, and the quantized model is deployed on the neural network processor (KPU) of the AI chip of the developed system. Limited TensorFlow operators are supported by the KPU, and these operators are present in the conventional neural network layer. The constructed YOLO-LW model can extract the bounding boxes of the human and bed and the first-stage posture.

D. Posture Correction

Since the first-stage posture did not consider the relative positions of the subject and the bed, it might cause the error of posture classification. Therefore, the relative positions of the subject and the bed is also considered to correct the first-stage posture and enhance the reliability of posture classification. After obtaining the bounding boxes of the subject and the bed, the distance between the human body and the edge of the bed is set as a feature to judge whether the subject's body is inside or outside the bed. To enable the thermopile array sensing element to capture the bed position, USB heating pads (TSA1000125eR1.06, Taiwan KLC PTC Co. Ltd., Taichung, Taiwan) are installed on the four corners of the bed used in the present study. This design allows the proposed system to adapt to beds of different sizes without any calibration and correct the first-stage posture. Fig. 6(a) and (b) display the bounding boxes captured by the AI model of the developed system when the human is outside and inside the bed, respectively.

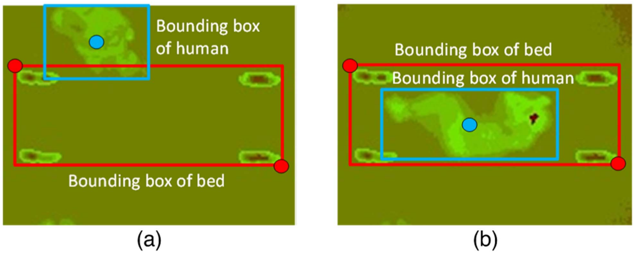

Fig. 6. Bounding boxes produced by the AI model of the developed system when the user is (a) outside and (b) inside the bed.

The bounding box of the bed can be used to calculate the squared y-axis distance (D^2) between the center of the bounding box of the human and the edge of the bed (11). When $D^2 > 0$, the user is outside the bed; when $D^2 < 0$, the user is inside the bed.

$$D^2 = \left(y_C^H - y_{TL}^B\right)\left(y_C^H - y_{BR}^B\right) \tag{11}$$

where y_C^H represents the y-coordinate of the center point of bounding box of human. y_{TL}^B represents the y-coordinate of the top left point of bounding box of bed. y_{BR}^B is the y-coordinate of the bottom right point of bounding box of bed. The (x, y) coordinates of the upper left and that of the bottom right corner in the thermal image are (0, 0) and (159, 119), respectively.

In addition to use the relative positions of the subject and the bed to correct posture, the classification error caused by the problem of unstable posture recognition result is also need to be addressed. Therefore, a sliding window with the size of 5 samples is adopted to detect the fall event [22]. The fall event is identified when the falling state is detected more than three times in a window.

E. Experimental Setting and Procedures

In this study, data were collected on 10 participants, namely five men and five women, with an average age of 49.9 ± 20.7 years. The experimental procedure for data collection was approved by the Institutional Review Board of En Chu Kong Hospital, New Taipei City, Taiwan (IRB number: ECKIRB1101201). The thermopile array sensing element of the developed system was installed on the ceiling directly above the bed, as shown in Fig. 7, to prevents artificial occlusion during imaging. The distance between the sensing element and the ground was 3 m, and the height of the bed was 0.5 m; thus, the sensor was located 2.5 m away from the bed surface. The obtained data set contained 10802 images of standing (2702 photos), sitting (2698 photos), falling (2699 photos of 132 falls), and nine sleeping postures (2703). The images of the 12 postures are depicted in Fig. 8. The images of sleeping postures comprised 296, 308, 295, 304, 301, 297, 299, 303, and 300 images of lying upright, lying in a star shape, lying on the stomach, lying on the left side, lying on the left side with arms open, lying on the left side with a curved body, lying on the right side, lying on the right side with arms open, and lying on the right side with a curved body, respectively.

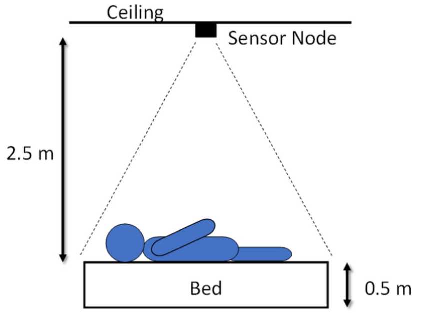

Fig. 7. Installation condition for the thermopile array sensing element.

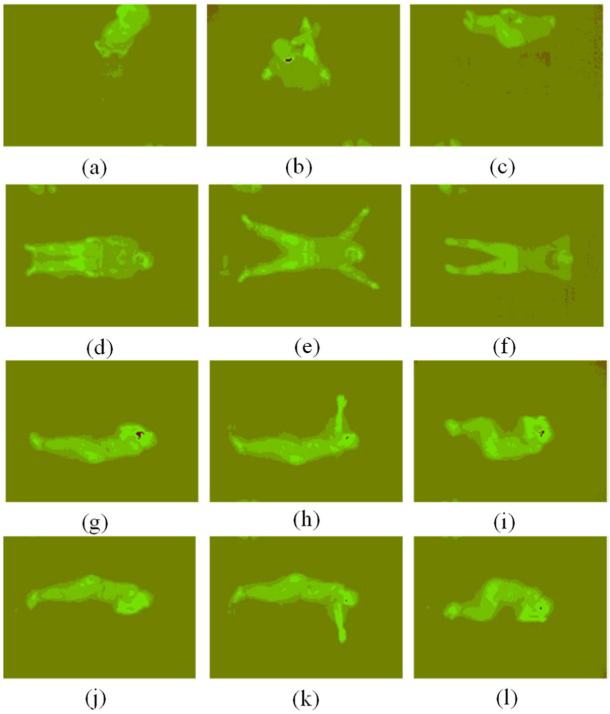

Fig. 8. Different human postures: (a) Standing, (b) sitting, (c) falling, (d) lying upright, (e) lying in a star shape, (f) lying on the stomach, (g) lying on the left side, (h) lying on the left side with arms open, (i) lying on the left side with a curved body, (j) lying on the right side, (k) lying on the right side with arms open, and (l) lying on the right side with a curved body.

Intersection over Union (IoU) was used to evaluate the boundary detection performance of the adopted neural network. IoU is calculated as follows:

$$IoU = \frac{O_{GP}}{U_{GP}} \tag{12}$$

where O_{GP} denotes the overlapping area between the predicted bounding box and the ground truth, and U_{GP} denotes the union of the area of the ground truth and that of the predicted bounding box.

The overall performance of the model is evaluated with five-fold cross-validation. The indicators including $F1$-score, accuracy, recall, precision, and specificity, and were adopted to evaluate the performance of the classifier and overall system. To increase the inferencing speed of the developed system, simulations were performed on a high-performance PC with a GPU, and it was converted into a k model that could be executed by the AI board after finding a suitable AI model. The experimental process to evaluate the performance of different models on the PC and AI board is illustrated in Fig. 9.

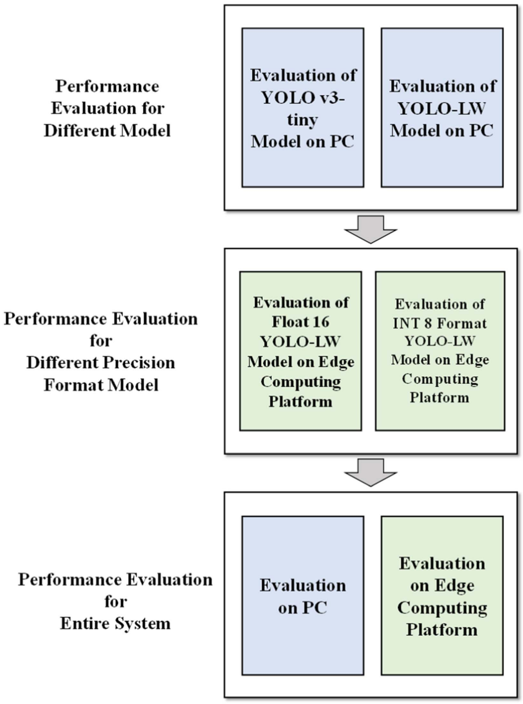
Performance Evaluation for Different Model
Evaluation of YOLO v3-tiny Model on PC
Evaluation of YOLO-LW Model on PC
Performance Evaluation for Different Precision Format Model
Evaluation of Float 16 YOLO-LW Model on Edge Computing Platform
Evaluation of INT 8 Format YOLO-LW Model on Edge Computing Platform
Performance Evaluation for Entire System
Evaluation on PC
Evaluation on Edge Computing Platform

Fig. 9. The experimental process for evaluating the performance of different models on the PC and AI board.

TABLE I
IoU Values Obtained Using YOLO v3-Tiny and YOLO-LW

Iteration	YOLO v3-tiny (%)	YOLO-LW (%)
1	96.45	95.64
2	96.71	95.44
3	95.89	94.86
4	95.63	95.63
5	96.16	95.72
Average	96.16	95.45

III. Results

A. Comparison Between the Original YOLO V3-Tiny and YOLO-LW Models

The anchors in the training dataset was found using K-means. The learning rate was 0.001. The training process finished when the loss of the model was lower than 0.05.

Table I presents the values of IoU obtained for human body recognition by using the YOLO v3-tiny and YOLO-LW models with the 32-bit floating-point format. Tables II and III present the performance indicators of the aforementioned models, respectively, when using them for sleeping posture classification. These indicators were determined to compare the performance of the YOLO-LW model on the PC with that of the YOLO v3-tiny model and to determine whether the YOLO-LW model

TABLE II
PERFORMANCE OF YOLO V3-TINY IN CLASSIFICATION OF FIRST-STAGE POSTURE

Iteration	Acc. (%)	Prec. (%)	Rec. (%)	Spec. (%)	F1-score (%)
1	96.82	96.31	97.18	96.48	96.74
2	96.34	96.61	95.81	96.83	96.20
3	95.65	95.4	96.05	95.22	95.72
4	95.24	95.44	94.94	95.53	95.18
5	95.27	94.93	95.52	95.01	95.23
Average	95.86	95.74	95.9	95.81	95.81

TABLE III
PERFORMANCE OF YOLO-LW IN CLASSIFICATION OF FIRST-STAGE POSTURE

Iteration	Acc. (%)	Prec. (%)	Rec. (%)	Spec. (%)	F1-score (%)
1	93.42	93.83	93.09	95.76	95.46
2	93.75	94.01	93.54	95.96	95.79
3	94.63	94.48	94.73	94.52	94.6
4	94.01	95.19	94.88	94.86	95.03
5	94.18	94.13	93.94	94.4	94.04
Average	93.99	94.32	94.03	94.3	94.18

fulfilled the preliminary requirements. Because YOLO-LW fulfilled these requirements, the performance of the entire developed system was evaluated.

As presented in Table I, the original YOLO v3-tiny model detected all human bodies in all thermal images, and its average IoU was 96.16%, even under the influence of residual temperature. Because YOLO-LW uses depth-wise separable convolutional layers instead of general convolutional layers (as in YOLO v3-tiny) to reduce model parameters, its average IoU (95.45%) was marginally lower than that of YOLO v3-tiny. Thus, YOLO-LW was more computationally efficient than YOLO v3-tiny. The IoU of the bounding box of the bed detected by YOLO-LW was 94.76%.

As presented in Table II, in the classification of first-stage postures, the average accuracy of YOLO v3-tiny was 95.86%. The average values of this model's other performance indicators also exceeded 95%.

As presented in Table III, in the classification of first-stage postures, the average accuracy of YOLO-LW was 93.99%, and the average values of this model's other performance indicators exceeded 94%.

B. Experimental Results for Models With Different Precision Formats

To estimate the effect of quantization on YOLO-LW, two models with different precisions, namely the float 16 and int 8 models, were tested. Because the KPU only supports the inferencing of the int 8 model, the float 16 model was inferenced by using the CPU on the edge computing platform, whereas the int 8 model was inferenced by the KPU. The relevant test results are shown in Table IV. According to this table, the average IoU and average posture classification accuracy of the float 16 model were 94.32% and 93.86%, respectively. Moreover, the corresponding values for the int 8 format model were 93.41% and 92.55%, respectively. The inferencing speeds of the float 16 and int 8 models were 0.69 and 5.28 frames per second (FPS), respectively. Considering the model inferencing speed required in emergency processing, the int 8 model was selected for subsequent evaluation.

TABLE IV
PERFORMANCE INDICATORS OF FLOAT 16 AND INT 8 MODELS IN CLASSIFICATION OF FIRST-STAGE POSTURES

Iteration (Average)	float 16 (CPU on Edge Computing Platform)	int 8 (KPU on Edge Computing Platform)
IoU	94.32	93.41
Acc.	93.86	92.55
Prec.	94.05	93.24
Rec.	93.92	93.02
Spec.	94.02	93.14
F1-score	93.91	92.71
FPS	0.69	5.28
Size (MB)	6.34	3.58

TABLE V
PERFORMANCE INDICATORS OF DIFFERENT AI MODELS IN CLASSIFICATION OF FINAL-STAGE POSTURE ON PC AND EDGE COMPUTING PLATFORM

	Acc. (%)	Prec. (%)	Rec. (%)	Spec. (%)	F1-score (%)
YOLO v3-tiny + Posture Classification Correction (on PC)	96.31	96.48	96.15	96.46	96.3
YOLO-LW + Posture Classification Correction (on PC)	95.23	94.9	95.6	94.85	95.22
YOLO-LW + Posture Classification Correction (on Edge Computing Platform)	94.56	94.52	94.33	94.77	94.54

C. Overall System Evaluation

The accuracy of human body boundary detection and posture classification when using the developed system was examined. Table IV presents the performance of executing the float 16 and int 8 models on the edge computing platform. The results indicate that the int 8 model achieved an average IoU, classification accuracy, and inferencing speed of 93.41%, 92.55%, and 5.28 FPS on the edge computing platform, respectively. The first-stage posture and human position feature (D^2) obtained using the AI model were used to determine final-stage posture. If the AI classification results indicated that the user was in a sleeping posture (first-stage posture) outside the bed, the classification result was modified to falling. Table V presents the performance of using different AI models in classification of final-stage posture on edge computing platform and PC. When YOLO v3-tiny is used, the overall accuracy of the system yields 96.31% on the PC. When some of the convolutional layers of YOLO v3-tiny were replaced with depth-wise separable convolutional layers, the system accuracy decreased to 95.23%. When YOLO-LW was converted to the int 8 precision format, partial information was lost, which led to decreases in the IoU and classification accuracy (the classification accuracy decreased to 94.56%).

IV. Discussion

This study developed a bedside fall monitoring system containing a thermopile array sensing element and a neuromorphic computing hardware. Neuromorphic computing hardware was used instead of PC to implement deep learning for the falls detection. The YOLO-LW model was deployed on the edge computing platform to infer neural networks in the int 8 precision format. With this design, the execution time can be reduced without decreasing the accuracy of posture classification.

Various thermal images collected in indoor were input into YOLO v3-tiny and three YOLO-LW models with different precision formats, and the classification performance of these models on a PC was compared by performing five-fold cross-validation. According to Tables I and II, the average IoU of YOLO v3-tiny reached 96.16%, and each performance indicator of this model for posture classification exceeded 95%. However, the weight of YOLO v3-tiny was too high for running it on the Sipeed Maix GO edge computing platform. YOLO-LW is designed for deployment on edge computing devices. According to Tables I and III, the average IoU and average classification accuracy of YOLO-LW were 0.71% and 1.87% lower than those of YOLO v3-tiny, respectively. However, YOLO-LW uses nearly half the parameters that YOLO v3-tiny does; thus, the size of YOLO-LW (12.68 MB) was considerably smaller than that of YOLO v3-tiny (33.7 MB). Studies have indicated that convert the floating-point format from a high-precision to a low-precision format reduce the model size effectively while maintaining network accuracy [35], [36]. Table IV presents the performance of the float 16 and int 8 models in the present study. Compared with the float 32 YOLO-LW model implemented on the PC, the float 16 and int 8 YOLO-LW models used 50% and 75% fewer parameters, respectively, had a 1.13% and 2.04% lower average IoU, respectively, and exhibited a 0.13% and 1.44% lower average classification accuracy, respectively. The experimental results indicated that the IoU and classification accuracy decreased marginally and notably, respectively, with a decrease in the precision level of YOLO-LW. The main reason for these results is that some falling actions were misclassified as bed postures. Because the KPU used in the developed system does not support the float 16 model, a CPU was used for inferencing this model, which achieved an inferencing speed of only 0.69 FPS. Because of the use of a lightweight model (YOLO-LW), the prior knowledge of neurons (i.e., zero point and scale factor), and dedicated hardware (i.e., a KPU), the int 8 model achieved an inferencing speed of 5.28 FPS.

Table V presents the performance of the entire developed system on the adopted PC and edge computing device. The overall system used a GPU for inferencing on the PC. The average classification accuracy of the overall system on the PC was 96.31% when this system contained YOLO v3-tiny. This classification accuracy was 0.45% higher than that achieved when using YOLO v3-tiny alone. Moreover, the overall system achieved an average accuracy of 95.23% when it contained YOLO-LW. This accuracy was 1.24% higher than that achieved when using YOLO-LW alone. The main reason for the aforementioned results is that the overall system can distinguish falling and sleeping postures on the basis of bed position. The IoU of YOLO-LW was lower than that of YOLO v3-tiny; thus, the bed positioning accuracy of YOLO-LW was lower than that of YOLO v3-tiny. Consequently, the overall system had a 1.08% lower average classification accuracy when it contained YOLO-LW than when it contained YOLO v3-tiny.

According to Table V, when the overall system that contained YOLO-LW and posture classification correction was implemented on an edge computing device, the average classification accuracy of this system (94.56%) was 2.01% higher than that achieved when using YOLO-LW alone. This result was obtained because of identifying bed position can effectively help system to execute posture classification correction to improve the overall system performance. Users can choose not to use the heating pads to get slightly worse but acceptable use (i.e., first-stage posture, accuracy = 92.55%) or to use the heating pads and posture classification correction to achieve a better performance (i.e., final-stage posture, accuracy = 94.56%). Table VI presents a comparison between the proposed method and other methods for bedside monitoring. Zhao et al. proposed a method based on the random forest (RF) algorithm [16]. In this method, the data captured by a depth camera are systematically analyzed to identify bedside falls. Although this method can achieve high accuracy (92.3%), it requires high network bandwidth because images are transmitted to a back-end PC. Chen et al. used histograms of oriented gradients (HOG) and principal component analysis (PCA) to filter the features, and then used a multilayer perceptron (MLP) to classify nine postures with an accuracy of 99.8% but could only detect in-bed postures [17]. Asbjørn et al. used a two-stage k-nearest neighbors (KNN) method to analyze the data captured by thermopile array sensing elements and ultrasonic data to detect bedside fall events [20]. Wickramasinghe et al. combined radio-frequency identification (RFID) tag and a three-axis accelerometer to analyze bed-egress movement [37]. However, it only detected the in-bed and out-bed events but did not detect the other sleeping postures. Guo et al. is the most similar study compared to the methods proposed in this paper. Guo et al. used cameras to detect sleeping postures and fall events with convolutional neural networks (CNN) and attention mechanism [38]. However, the accuracy of study is lower than the accuracy of our proposed method. Moreover, using a PC with GPU requires higher power consumption. Compared to the related works, the edge computing method proposed in this paper is faster than the methods proposed in [16], [17], [20], [37] and [38] and can detect in-bed postures and out-of-bed falls. The inferencing speed of the developed system is 5.28 FPS, which can meet the requirements of practical use. Regarding the power consumption, the thermopile array sensing element used as the input sensor in the developed system is more power-efficient than a depth camera. Moreover, the AI board of this system has low power consumption (1.1 W), low cost, and low computational complexity, especially compared to the research

TABLE VI
PARAMETERS OF DIFFERENT AI MODELS AND PLATFORMS IN FINAL-STAGE POSTURE CLASSIFICATION

	Proposed Method	Zhao et al. [16]	Chen et al. [17]	Asbjørn et al. [20]	Wickramasinghe et al. [37]	Guo et al. [38]
Core	Sipeed Maix Go	PC with Intel Core I7-6700K CPU	PC	Beaglebone Black	PC	PC with GeForce RTX 3090
Power Consumption of the Platform	1.5 W	Over 83 W	-	2.3 W	-	Over 350 W
Sensor	Thermopile Array	Single Depth Camera	Thermopile Array	Thermopile Array + Ultrasonic	RFID + Accelerometer	Camera
Algorithm	YOLO-LW + Correction	Mean-Shift Clustering + RF	HOG + PCA + MLP	Two-Stage KNN	Support Vector Machines	CNN + Attention Mechanism
FPS	5.28	-	-	0.1	0.2	-
Performance of Sleeping Posture Recognition	Accuracy: 94.56% Precision: 94.52% Recall: 94.33% Specificity: 94.77% F1-score: 94.54%	-	Accuracy: 99.8%	-	-	Accuracy: 90.67%
Performance of Fall Event Detection	Accuracy: 94.56% Precision: 94.52% Recall: 94.33% Specificity: 94.77% F1-score: 94.54%	Accuracy: 92.3% Precision: 97.8% Recall: 86.6% Specificity: 98.1%	-	Accuracy: 98.62% Precision: 92.86% Recall: 100% Specificity: 98.32%	Accuracy: 96.1% Recall: 96.3% Specificity: 96.2%	Accuracy: 90.67%
System Is Not Easily Affected by Light	Yes	No	Yes	Yes	Yes	No
Edge Computing	Yes	No	No	Yes	No	No
Bedside Fall Event Detection	Yes	Yes	No	Yes	Yes	Yes
Automatic Bounding of Bed	Yes	No	No	No	No	No
Sleeping Posture Recognition	Yes	No	Yes	No	No	Yes

using PCs. However, the thermopile array sensing element used in this study has certain limitations. When the user is covered with a thick quilt, their body temperature cannot be sensed accurately, which results in an inability to recognize postures on the bed. Fortunately, the developed system can recognize the actions of sitting, standing, and falling and can provide the function of automatically turning on the lights for the elderly when they get out of bed, thereby providing emergency aid for falling.

V. CONCLUSION

This article proposed a bedside fall and sleep posture monitoring system based on edge computing and neuromorphic computing hardware. The AI model of the proposed system firstly outputs the first-stage posture, and bounding boxes of the individual and the bed. The bounding boxes are used to calculate the feature, which is used to determine the position of the user. This position determines the classification of the user's final sleeping posture (12 types of postures). In this study, the YOLO-LW neural network model was successfully deployed on an edge computing platform. In contrast to models with the float 32 precision format that are used on a PC, YOLO-LW uses depth-wise separable convolutional layers to improve its computational efficiency. The floating point in YOLO-LW was converted into int 8 precision format to speed up inferencing and reduce size of the model. Moreover, additional convolutional layers were added to this model to maintain its accuracy. The average accuracy of the system is 94.56%. The inferencing speed of the system is 5.28 FPS. The power consumption of the edge computing platform with the thermopile array sensing element is 1.5 W. All the computations were conducted on the edge computing platform, and only the posture classification was transmitted to the back end, thereby protecting user privacy. However, the developed system requires the sensing element to be installed directly above the bed, which might cause user unease. In the future, different capture angles can be evaluated to identify a suitable capture angle for reducing user unease.

References

[1] S. Naja, M. M. E. D. Makhlouf, and M. A. H. Chehab, "An ageing world of the 21st century: A literature review," *Int. J. Community Med. Public Health*, vol. 4, no. 12, pp. 4363–4369, Nov. 2017.

[2] D. C. Ranasinghe, R. L. S. Torres, K. Hill, and R. Visvanathan, "Low cost and batteryless sensor-enabled radio frequency identification tag based approaches to identify patient bed entry and exit posture transitions," *Gait Posture*, vol. 39, no. 1, pp. 118–123, Jan. 2014.

[3] D. A. Sterling, J. A. O'Connor, and J. Bonadies, "Geriatric falls: Injury severity is high and disproportionate to mechanism," *J. Trauma Acute Care Surg.*, vol. 50, no. 1, pp. 116–119, Jan. 2001.

[4] M. P. Aust, "Pressure ulcer prevention," *Amer. J. Crit. Care*, vol. 20, no. 5, p. 376, Sep. 2011.

[5] W.-Y. Lin, C.-H. Chen, and M.-Y. Lee, "Design and implementation of a wearable accelerometer-based motion/tilt sensing internet of things module, and its application to bed fall prevention," *Biosensors*, vol. 11, no. 11, Oct. 2021, Art. no. 428.

[6] N.-J. Raden, U. Kulau, M. Marschollek, and K.-H. Wolf, "INBED: A highly specialized system for bed-exit-detection and fall prevention on a geriatric ward," *Sensors*, vol. 19, no. 5, Feb. 2019, Art. no. 1017.

[7] M. Awais et al., "An internet of things based bed-egress alerting paradigm using wearable sensors in elderly care environment," *Sensors*, vol. 19, no. 11, May 2019, Art. no. 2498.

[8] G. Matar, J.-M. Lina, and G. Kaddoum, "Artificial neural network for in-bed posture classification using bed-sheet pressure sensors," *IEEE J Biomed. Health Inform.*, vol. 24, no. 1, pp. 101–110, Jan. 2020.

[9] G. Matar, G. Kaddoum, J. Carrier, and J.-M. Lina, "Kalman filtering for posture-adaptive in-bed breathing rate monitoring using bed-sheet pressure sensors," *IEEE Sensors J.*, vol. 21, no. 13, pp. 14339–14351, Jul. 2021.

[10] Q. Hu, X. Tang, and W. Tang, "A real-time patient-specific sleeping posture recognition system using pressure sensitive conductive sheet and transfer learning," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6869–6879, Mar. 2021.

[11] Y. Gu, Y. Wang, Z. Liu, J. Liu, and J. Li, "SleepGuardian: An RF-based healthcare system guarding your sleep from afar," *IEEE Netw.*, vol. 34, no. 2, pp. 164–171, Mar./Apr. 2020.

[12] P.-J. Chen, T.-H. Hu, and M.-S. Wang, "Raspberry Pi-based sleep posture recognition system using AIoT technique," *Healthcare*, vol. 10, no. 3, Mar. 2022, Art. no. 513.

[13] X. Yang et al., "Monitoring of patients suffering from REM sleep behavior disorder," *IEEE J. Electromagn. RF Microw. Med. Biol.*, vol. 2, no. 2, pp. 138–143, Jun. 2018.

[14] K.-L. Chung, L.-T. Liu, and C.-H. Liao, "Novel and robust vision- and system-on-chip-based sensor for fall detection," *Sensors Mater.*, vol. 31, no. 8, pp. 2657–2668, Aug. 2019.

[15] F. Deng et al., "Design and implementation of a noncontact sleep monitoring system using infrared cameras and motion sensor," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 7, pp. 1555–1563, Jul. 2018.

[16] F. Zhao, Z. Cao, Y. Xiao, J. Mao, and J. Yuan, "Real-time detection of fall from bed using a single depth camera," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 3, pp. 1018–1032, Jul. 2019.

[17] Z. Chen and Y. Wang, "Remote recognition of in-bed postures using a thermopile array sensor with machine learning," *IEEE Sensors J.*, vol. 21, no. 9, pp. 10428–10436, May 2021.

[18] S. Transue, P. Nguyen, T. Vu, and M.-H. Choi, "Volumetric reconstruction of thermal-depth fused 3D models for occluded body posture estimation," *Smart Health*, vol. 11, pp. 29–44, Jan. 2019.

[19] K. Naik, T. Pandit, N. Naik, and P. Shah, "Activity recognition in residential spaces with internet of things devices and thermal imaging," *Sensors*, vol. 21, no. 3, Feb. 2021, Art. no. 988.

[20] D. Asbjørn and T. Jim, "Recognizing bedside events using thermal and ultrasonic readings," *Sensors*, vol. 17, no. 6, Jun. 2017, Art. no. 1342.

[21] W.-J. Chang, C.-H. Hsu, and L.-B. Chen, "A pose estimation-based fall detection methodology using artificial intelligence edge computing," *IEEE Access*, vol. 9, pp. 129965–129976, 2021.

[22] B.-S. Lin et al., "Fall detection system with artificial intelligence-based edge computing," *IEEE Access*, vol. 10, pp. 4328–4339, 2022.

[23] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Mini-YOLOv3: Real-time object detector for embedded applications," *IEEE Access*, vol. 7, pp. 133529–133538, 2019.

[24] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020.

[25] W.-J. Chang, C.-H. Hsu, and L.-B. Chen, "A pose estimation-based fall detection methodology using artificial intelligence edge computing," *IEEE Access*, vol. 9, pp. 129965–129976, 2021.

[26] X. Liu et al., "Collaborative edge computing with FPGA-based CNN accelerators for energy-efficient and time-aware face tracking system," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 1, pp. 252–266, Feb. 2022.

[27] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. Operating Syst. Des. Implementation*, 2016, vol. 16, pp. 265–283.

[28] P. Gysel, J. Pimentel, M. Motamedi, and S. Ghiasi, "Ristretto: A framework for empirical study of resource-efficient inference in convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5784–5789, Nov. 2018.

[29] S. Carata, R. Mihaescu, E. Barnoviciu, M. Chindea, M. Ghenescu, and V. Ghenescu, "Complete visualisation, network modeling and training, web based tool, for the YOLO deep neural network model in the Darknet framework," in *Proc. IEEE 15th Int. Conf. Intell. Comput. Commun. Process.*, 2019, pp. 517–523.

[30] J. Redmon and A. Farhadi, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[31] G. Tian, J. Chen, X. Zeng, and Y. Liu, "Pruning by training: A novel deep neural network compression framework for image processing," *IEEE Signal Process. Lett.*, vol. 28, pp. 344–348, 2021.

[32] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, Apr. 2017.

[33] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4133–4141.

[34] S. Kim, G. Park, and Y. Yi, "Performance evaluation of INT8 quantized inference on mobile GPUs," *IEEE Access*, vol. 9, pp. 164245–164255, 2021.

[35] P. E. Novac, G. B. Hacene, A. Pegatoquet, B. Miramond, and V. Gripon, "Quantization and deployment of deep neural networks on microcontrollers," *Sensors*, vol. 21, no. 9, Apr. 2021, Art. no. 2984.

[36] P. Gysel, J. Pimentel, M. Motamedi, and S. Ghiasi, "Ristretto: A framework for empirical study of resource-efficient inference in convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5784–5789, Nov. 2018.

[37] A. Wickramasinghe, D. C. Ranasinghe, C. Fumeaux, K. D. Hill, and R. Visvanathan, "Sequence learning with passive RFID sensors for real-time bed-egress recognition in older people," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 4, pp. 917–929, Jul. 2017.

[38] R. Guo, C. Zhai, L. Zheng, and L. Zhang, "Sleep behavior detection based on pseudo-3D convolutional neural network and attention mechanism," *IEEE Access*, vol. 10, pp. 90101–90110, 2022.