

Received December 13, 2021, accepted December 30, 2021, date of publication January 4, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3140164

# Fall Detection System With Artificial Intelligence-Based Edge Computing

**BOR-SHING LIN**<sup>1</sup>, (Senior Member, IEEE), **TIKU YU**<sup>2</sup>, **CHIH-WEI PENG**<sup>3</sup>,  
**CHUEH-HO LIN**<sup>4,5</sup>, **HUNG-KAI HSU**<sup>1</sup>, **I-JUNG LEE**<sup>1,6</sup>,  
**AND ZHAO ZHANG**<sup>1,6,7</sup>, (Graduate Student Member, IEEE)

<sup>1</sup>Department of Computer Science and Information Engineering, National Taipei University, New Taipei City 237303, Taiwan

<sup>2</sup>Department of Communication Engineering, National Taipei University, New Taipei City 237303, Taiwan

<sup>3</sup>School of Biomedical Engineering, College of Biomedical Engineering, Taipei Medical University, Taipei 11031, Taiwan

<sup>4</sup>Master Program in Long-Term Care, College of Nursing, Taipei Medical University, Taipei 11031, Taiwan

<sup>5</sup>Center for Nursing and Healthcare Research in Clinical Practice Application, Wan Fang Hospital, Taipei Medical University, Taipei 11031, Taiwan

<sup>6</sup>College of Electrical Engineering and Computer Science, National Taipei University, New Taipei City 237303, Taiwan

<sup>7</sup>School of Mechanical and Electrical Engineering, Wuyi University, Wuyishan, Fujian 354300, China

Corresponding author: Chueh-Ho Lin (chueh.ho@tmu.edu.tw)

This work was supported in part by the Ministry of Science and Technology in Taiwan under Grant MOST 110-2314-B-305-001 and Grant MOST 109-2221-E-305-001-MY2; in part by the University System of Taipei Joint Research Program under Grant USTP-NTPU-TMU-107-02, Grant USTP-NTPU-TMU-108-01, Grant USTP-NTPU-TMU-109-03, and Grant USTP-NTPU-NTOU-110-01; in part by the Faculty Group Research Funding Sponsorship by National Taipei University under Grant 2021-NTPU-ORDA-02; and in part by the National Taipei University, Taiwan, through the Academic Top-Notch and Features Field Project under Grant 110-NTPU\_ORDA-F-003.

**ABSTRACT** Falls are the second leading cause of death from unintentional injuries in older adults. Although many systems have been used to detect falls, they are limited by the computational complexity of the algorithm. The images taken by the camera must be transmitted through a network to the back-end server for calculation. As the demand for Internet of Things increases, this architecture faces such problems as high bandwidth costs and server computing overload. Emerging methods reduce the workload of servers by transferring certain computing tasks from cloud servers to edge computing platforms. To this end, this study developed a fall detection system based on neuromorphic computing hardware, which streamlines and transplants the neural network model of the back-end computer to the edge computing platform. Through the neural network model with integer 8 bit precision deployed on the edge computing platform, the object photos obtained by the camera are converted into human motion features, and a support vector machine is then used for classification. After experimental evaluation, an accuracy of 96% was reached, the detection speed of the overall system was 11.5 frames per second, and the power consumption was 0.3 W. This system can monitor the fall events of older adults in real time and over a long period. All data were calculated on the edge computing platform. The system only reports fall events via Wi-Fi, thereby protecting the privacy of the user.

**INDEX TERMS** Deep learning, edge computing, fall detection, neuromorphic computing hardware, the IoT.

## I. INTRODUCTION

The older adult population is expected to reach 1.4 billion by 2030 and 2.1 billion by 2050 [1]. With age, older adults experience more impairment in vision, balance, and cognition, all of which increase the chances of a fall. Thirty percent of elderly people over 65 years fall at least once every year, causing severe or even fatal damage. However, only one-third of people received medical assistance following a fall. The

The associate editor coordinating the review of this manuscript and approving it for publication was György Eigner<sup>1</sup>.

medical cost of fatal older adult falls was an estimated US 754 million dollars in 2015 [2].

In traditional fall detection systems for older adults, sensors and cameras are used to track the motion of individuals, and the sensor data and image data are sent to servers for analysis [3]–[9]. When a fall event is detected by the system, the server immediately notifies medical staff of the emergency. In research that employs cloud analysis, data can be analyzed using relatively sufficient and powerful back-end equipment to achieve higher accuracy. However, the main disadvantage of uploading a large amount of data to

the cloud server is the resultant high cost in network bandwidth, high latency, and privacy concerns [10]. With too many users, the network bandwidth and loading of the cloud computing may become unfeasible. The rapid advancement of chip technology resulted in algorithms that could not be calculated in real time on the front-end being transferred from the cloud to the front-end, reducing the burden on the server. An edge computing-based system could also be used to perform better in real time than a cloud-based system [11], which is highly valuable for fall detection systems. Therefore, some fall detection systems based on edge computing have been proposed, including architectures using general-purpose processors [12]–[17] and neuromorphic computing hardware [18]. The general-purpose processor architecture used in many studies is relatively easy to implement but consumes considerable time and computing resources to complete the neural network algorithm. These studies have been limited to the use of relatively low-rank robust methods such as statistics and thresholds. Although [17] used deep learning methods in which continuous frames were input to the neural network to train it to dynamically recognize falls, the system was forced to reduce the image resolution to  $32 \times 32$  to achieve real-time computation on the central processing unit (CPU). Low-resolution images can only provide close-range information, which limits their potential applicability. Other studies have demonstrated that a fall detection system based on deep learning can effectively improve detection accuracy [18]. However, if a fall detection system based on deep learning is to be implemented on an edge computing platform with general-purpose processor architecture, it may be more time-consuming and may not achieve real-time fall detection.

To address the aforementioned problems, this study proposed a fall detection system based on edge computing, which combined a camera and neuromorphic computing hardware based on an application-specific integrated circuit. The You Only Look Once lightweight (YOLO-LW) deep neural network was implemented on the neuromorphic computing hardware. Experiments have validated that the YOLO-LW algorithm combined with a support vector machine (SVM) can run smoothly on the edge computing platform and can accurately detect fall events in real time. In this study, the captured images are not uploaded to the cloud server, so when a large number of cameras are installed in practical applications, fall detection system does not occupy additional large amount of bandwidth, and server is not blocked by processing images from all cameras at the same time. And the edge computing platform sends a warning to server only if a fall event is detected, so the transmission delay can be negligible. Thus, user privacy is protected to a certain extent.

## II. RELATED WORKS

Several studies on fall detection have been proposed to reduce older adult fall injuries or provide emergency assistance after falls [7]–[9], [13], [14], [19], [20]. This section presents three categories of fall detection technologies: backend computing

fall detection, edge computing fall detection, and cloud-edge computing fall detection.

Harrou *et al.* [7] proposed an integrated vision-based fall detection approach implemented on a backend computer. The integrated vision-based fall detection approach involves image processing (background subtraction), morphological processing (erosion and dilation operators), centroid calculation, generalized likelihood ratio (GLR) calculation, and SVM. Image processing is used to segment the human body from the picture of the University of Rzeszow (UR) fall detection dataset. The human body contour obtained through image processing is divided into five areas. The five areas are passed to GLR-SVM classifiers to distinguish between real falls and certain like-falls activities. The approach was designed to detect fall event with fewer false-positive.

Wang *et al.* [8] proposed a novel visual-based fall detection approach by dual-channel feature integration. The research combines traditional signal processing with deep learning model. YOLO and OpenPose were used to obtain position and key points of a human body. A dual-channel sliding window was designed to extract the features (centroid speed, upper limb velocity, and human external ellipse) from result of deep learning model. Multilayer perceptron (MLP) and random forest were used to classify feature data. Then, the results of the classifiers were combined to detect fall events. The proposed approach achieved an accuracy of 97.33% and 96.91% on UR dataset and Le2i dataset separately.

Lotfi *et al.* [9] also proposed a novel visual-based fall detection approach. Background subtraction is used for pre-processing. Ten features are extracted from the human silhouette, including the motion information, orientation, ratio, the major semi-axis and the minor semi-axis of the fitting ellipse, the projection histogram, the y-coordinate of the head point, the standard deviation of y-coordinate, the absolute difference of y-coordinate and the standard deviation of absolute difference of y-coordinate. These features are fed into MLP neural network for fall classification. The proposed algorithm produces a high recognition rate of 99.60% on UR dataset.

Some computationally intensive technologies including image processing, and deep learning are used in the aforementioned studies. These technologies will burden backend computer when the number of cameras is large. Because the image data need to be transmitted to the backend computer, may cause security and privacy issues. In addition, some algorithms require continuous high-resolution images. This means that a lot of bandwidth will be occupied, may result in packet blockages and losses due to insufficient bandwidth.

ShahZad and Kim [13] proposed a two-step algorithm to monitor and detect fall events using the embedded accelerometer signals. The fall detection system was developed on a smartphone. The smartphone is placed on the waist or leg. The accelerometer signals on smartphone are passed to two-step algorithm for fall classification. Two-step algorithm is combination of threshold-based method and multiple kernel learning SVM. Experimental results reveal that the system

detects falls with high accuracy 97.8% and 91.7% on the waist and the leg.

Saleh and Jeannès [14] proposed a machine learning-based fall detection algorithm designed to deploy on wearable sensor. 12-component statistical features vector is extracted from a triaxial acceleration signal. The SVM-based algorithm is used to classify fall events. The experimental results show that the proposed algorithm can reach 99.9% on Sisfall dataset. In addition, the system implements algorithm with a computational cost of less than 500 floating point operations per second.

Yu *et al.* proposed a fall detection system based on neuro-morphic computing hardware [19] in which the user wore a device with an embedded inertial measurement unit (IMU) to measure human movement and capture five types of activities, including falls.

The studies [13], [14], [19] implement fall detection algorithm on an embedded system and experimental results show high accuracy. These studies are limited to the use of lower computational cost methods such as SVM or threshold. In addition, wearable system is inconvenience.

Rajavel *et al.* [20] proposed an IoT-based healthcare smart system. The system detects fall events with cloud server, edge computing devices and cameras. The images captured from cameras are transmitted to edge computing devices via Wi-Fi. The edge computing devices filter the non-sensitive data to reduce the communication bandwidth between the edge layer and cloud layer by transmitting only the needful data to the cloud layer. A deep convolution neural network classifier is used to detect fall events on the cloud server. The system exploits only 150 kbps network bandwidth and 80 s response time compared to past research. In addition, the system spends 72.76 s on prediction and accuracy reaches 94.5%. The research [20] uses cloud-edge based computing framework to implement fall prediction and performance exceeds previous research. But the long prediction time and response time may cause a fall event that cannot be handled immediately in the actual situation.

In order to address the aforementioned problems, a fall detection method based on artificial intelligence (AI) edge computing with vision sensor was proposed in this study. The proposed algorithms are deployed on edge computing platforms with AI chips. All operations are done on the edge computing platform.

### III. METHOD

#### A. SYSTEM OVERVIEW

The main components of the edge computing platform used in this study were a camera (ov2640, OmniVision Technologies, Shanghai, China) and an AI development board (Sipeed MAix GO, Sipeed Technology, Shenzhen, China), as depicted in Fig. 1. The core chip Sipeed M1W of the AI development board was composed of Kendryte K210 (Canaan, Beijing, China) and the Wi-Fi chip ESP8285. The K210 has two RISC-V CPUs and a neural network processor (KPU) that

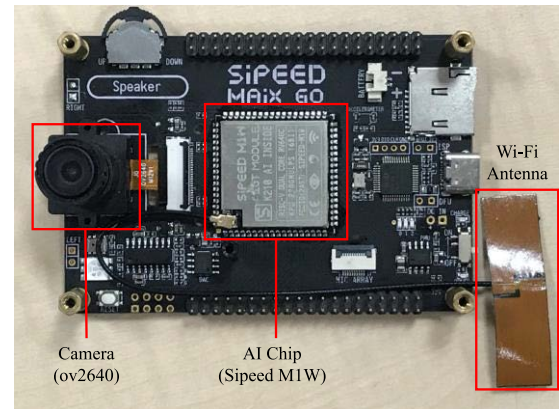


FIGURE 1. Prototype of the edge computing platform with an area of 60 × 88 mm.

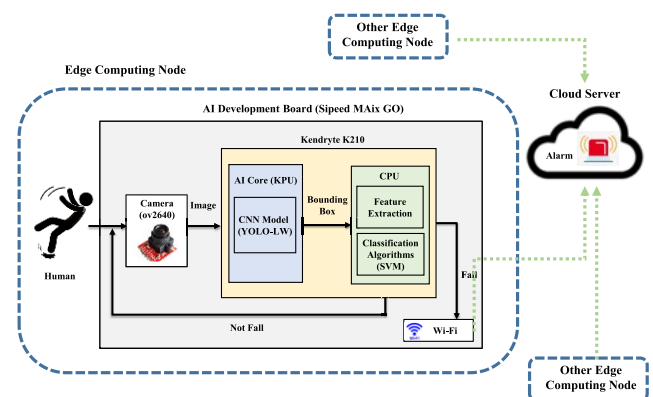
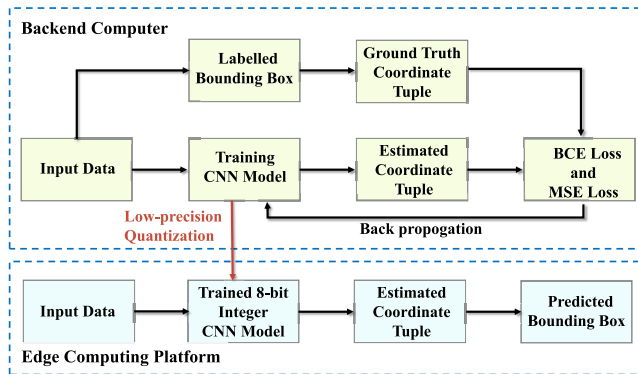


FIGURE 2. Flowchart of the proposed approach.

can perform AI operations, offering 0.25 tera operations per second (TOPS) at 0.3 W, 400 MHz and 0.5 TOPS when it overclocks to 800 MHz. The K210 can therefore perform object recognition at 60 frames per second (FPS) with a video graphics array. When the system is under operation, the edge computing platform continuously takes pictures through the camera, which are stored in the dedicated memory of the AI chip. The trained neural network structure deployed on the AI chip reads the data in the memory and calculates the bounding box of the human body in the image. If the human body is not captured in the image, no bounding box is computed. According to the bounding box, the shape aspect ratio (SAR) and the difference between width and height are obtained and used in the SVM classifier to classify actions, including standing, bending, and falling, in the CPU. When the state of human body changes from being at higher altitudes to fallen state and continues to be in that state for a period, it is determined a fall event. The system then transmits the fall event to the cloud server via Wi-Fi. The cloud server is only for monitoring and calling for emergency treatment. The entire process is illustrated in Fig. 2. The system reports the detection results and does not transmit images to protect the privacy of users.

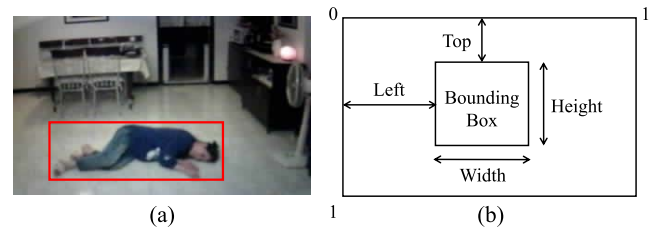
This study differs from those that have relied on cloud server computing [3], [4] or nondeep learning methods



**FIGURE 3.** Process of training and deploying neural networks.

[12], [13] to detect falls. In the AI model training phase, it is first trained using collected posture photos on the PC, and then the trained neural network on the edge computing platform is deployed. The PC specifications comprised a CPU (I7-10700F, Intel, Santa Clara, CA, USA), graphics processing unit (GPU) display card (RTX 3080 10G, AsusTeK, Taipei, Taiwan) and 64-gigabyte dynamic random-access memory. The neural network training framework Darknet [21] was used for training the AI model. The trained neural network is input to the conversion program and converted into a TensorFlow-based model that the KPU can infer. The KPU only partially supports TensorFlow operators, and the rebuilding of neural networks using supported operators to develop new edge computing neural networks is a gradual process.

Fig. 3 depicts the process of training and deploying neural networks. In preparation of the AI model training, the collected images were first labeled using free image labeling software (Labellmg) to manually generate the coordinate tuple data of the bounding box, including the width and height of the bounding box and the x- and y-axis distance from the upper left corner of the bounding box to the origin (upper left corner of the image). The coordinate tuple of the bounding box is used as the ground truth to train the neural network. Details of the labeled bounding box and coordinate tuple are presented in Fig. 4. During the following training process, the neural network makes inferences on the input images and generate estimated coordinate tuples. The estimated coordinate tuple and coordinate tuple of ground truth are then calculated for the loss function, and the neurons of the neural network are updated through back propagation. When the training is complete, low-precision quantization of the neural network is performed to increase the speed of the model and reduce the power consumption of the inferring model [22]–[24]. This low precision includes 16-bit float format, 8-bit integer format, and 4-bit integer format. The neural network on the computer side is converted from 32-bit float format to 8-bit integer format through low-precision quantization, during which the offset value in the neuron is ignored, and a threshold  $T$  is selected between the maximum



**FIGURE 4.** (a) Labeled bounding box; (b) Coordinate tuple of the ground truth: left, top, width, and height.

and minimum weight value in all neurons. The weight value between  $-T$ – $T$  is remapped to  $-127$  to  $127$ , and other weight values are discarded. To minimize the influence of the missing model information, the model is calibrated using the calibration dataset. The model uses different  $T$ s to infer the calibration dataset to determine the threshold  $T$  that least affects the accuracy of the model. This study employed the training dataset as the calibration data set, and the converted model was approximately one-fourth of the original model.

## B. DEVELOPMENT OF THE AI ALGORITHM AND CLASSIFIER ON EDGE AI BOARD

The most representative neural networks in object detection include single shot multibox detector [25], faster R-convolutional neural network (CNN) [26], You Only Look Once (YOLO) v2 [27], and YOLO v3 [28]. Compared with the traditional two-stage detection algorithm, YOLO v2 directly converts the bounding box positioning problem into an end-to-end regression solution and uses anchor boxes to detect objects of different sizes. The anchor box is determined by preset anchor points that represent the various proportions of the bounding box that may appear in the image. Because YOLO v2 avoids the process of generating hundreds of candidate boxes, the execution speed of the algorithm has markedly improved, ensuring the practical applicability of network. The YOLO v3 model was constructed in 2018 with the addition of multiscale prediction and a better basic classification network (i.e., Darknet-53). With faster speed and better accuracy in small target detection, detection distance can be extended in actual scenes. This study tested the lightweight models YOLO v2-tiny and YOLO v3-tiny on the PC by applying k-means clustering to group the training dataset. The center of each group represents the proportion of different bounding boxes in the training dataset. These values contain the proportional information of the human body set as anchor points, and the width and height of the neural network input layer is set as the maximum size limit size that the AI chip can process (width, 320 pixels; height, 224 pixels). The architectures of YOLO v2-tiny and YOLO v3-tiny are presented in Fig. 5 and Fig. 6, respectively. Although these methods achieve high accuracy on the PC, the model size is too large to be executed on the edge computing platform, which has limited memory. Therefore, a deep separable convolutional layer [29] is used to modify the neural network

and simplify the neurons. According to [29], the ratio of the amount of deep separable convolutional layers to that of the calculated ordinary convolutional layers can be expressed as (1).

$$\frac{Cost_{depth\_conv} + Cost_{point\_conv}}{Cost_{standard\_conv}} = \frac{1}{N} + \frac{1}{K^2} \quad (1)$$

where  $Cost_{depth\_conv}$  represents the computational cost of the deep convolutional layer;  $Cost_{point\_conv}$  represents that of the point convolutional layer;  $Cost_{standard\_conv}$  represents that of the ordinary convolutional layer;  $N$  represents the number of channels; and  $K$  denotes the size of the kernel.

Figs. 7 and 8 depict the modified YOLO v2-tiny and YOLO v3-tiny. According to Eq. (1), when the common convolutional layer is replaced by the deep separable convolutional layer with more filters (i.e., channels) and a larger kernel size, the computational cost is reduced. Considering the limitation of memory, we chose to replace the convolutional layer, with a filter amount greater than or equal to 512, with the deep separated convolutional layer. The filters in the output layer of YOLO v3-tiny are fewer than those in YOLO v2-tiny, and the calculation cost of the output layer is therefore reduced to approximately two-thirds that of YOLO v2-tiny. To further shrink YOLO v3-tiny, some convolutional layers with a filter amount of 256 were replaced with deep separated convolutional layers.

However, after low-precision quantization, the test indicated that the accuracy of the neural network executed on the edge computing platform decreased. To improve the accuracy, some convolutional layers were added to increase the depth of the neural network. Because the categories were changed from 85 of the original YOLO v3-tiny to 1 (only one category for humans), the filters of the output layer must be changed from the original 255 of the YOLO v3-tiny to 18. The addition of convolutional layers between the output layer and the previous layer and a decrease in filters better condenses the features, and the increased amount of calculation is sustainable. Eq. (2) expresses the relationship between the categories and the filters of output layer. Fig. 9 presents the architecture of the modified YOLO v3-tiny following the addition of the convolutional layers. In this study, this lightweight modified YOLO v3-tiny was named YOLO-LW.

$$F = (C + 5) \times A \quad (2)$$

where  $F$  is the number of filters in the output layer;  $C$  is the number of classes;  $A$  is the number of anchor points; and the number of YOLO v3-tiny anchor points is equal to 3. Therefore, the number of filters in the modified YOLO v3-tiny output layer is  $(1 + 5) \times 3 = 18$ .

## C. EXPERIMENTAL PROCEDURES

In this study, we collected data from 19 participants, including 11 men and 8 women. The ratio of young to old was 9:10, and their average age, height, and weight was  $46.3 \pm 16.1$  years,  $166.1 \pm 9.9$  cm, and  $67.3 \pm 12.8$  kg, respectively.

The collected data were captured in five different indoor environments. The camera was installed at a height of 1.7 m from the ground and a distance of 2–3.5 m from the object. The optical axis of the camera was tilted downwards at an angle of  $22.5^\circ$  to the horizontal. The camera recorded the fall process from eight different angles relative to the direction in which the participant fell. The data set contained a total of 2030 photos, consisting of 1077 falling and 953 nonfalling photos. This data set contained information on 152 falls and other actions such as walking, bending, squatting, sitting, and kneeling, and used horizontal flips to double the data.

As Fig. 2, the system obtains the images of the person through the camera. Next, the YOLO model is used to capture the silhouette of person, and then passing the extracted features from the silhouette of person to the SVM classifier. Results of SVM are classified into three classes: standing, blending, and falling. Finally, a sliding window is used to detect fall event.

This study employed intersection over union (IoU) to evaluate the neural network bounding performance, as calculated in (3).

$$IoU = \frac{\text{Overlap of Ground Truth and Predicted Bounding Box}}{\text{Union of Ground Truth and Predicted Bounding Box}} \quad (3)$$

Shape aspect ratio (SAR), and difference (D) between width and height, are extracted from the bounding box, the formulas of these two features are listed in (4) and (5).

$$SAR = \frac{\text{Width of Bounding Box}}{\text{Height of Bounding Box}} \quad (4)$$

$$D = \text{Width of Bounding Box} - \text{Height of Bounding Box} \quad (5)$$

A sliding window is designed to detect fall event. When the falling state appears more than three times in a sliding window, the result is considered a fall event.

To evaluate the performance of the classifier and the overall system, including indicators such as accuracy, precision, recall, specificity, and F1-score, 10-fold cross-validation was used. The indicator definitions are expressed in (6)–(10).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (9)$$

$$F1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

where true positive (TP) refers to the number of falls correctly detected; true negative (TN) refers to the number of normal activities that are correctly detected; false positive (FP) refers to the number of normal activities that are mistaken for

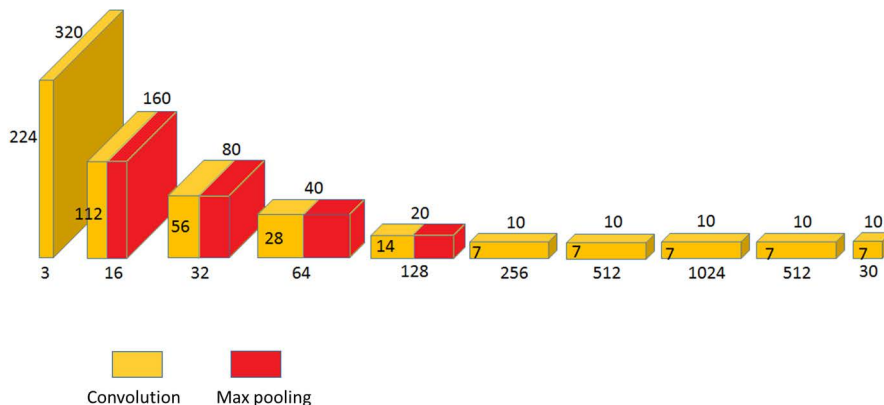


FIGURE 5. Architecture of YOLO v2-tiny.

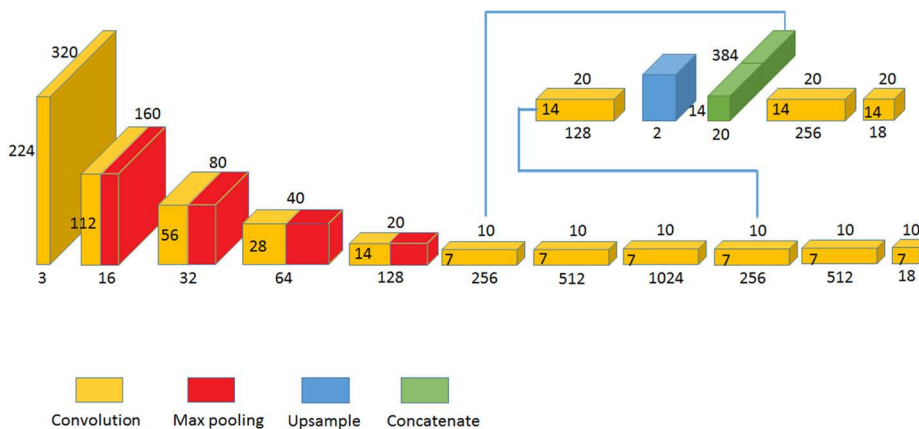


FIGURE 6. Architecture of YOLO v3-tiny.

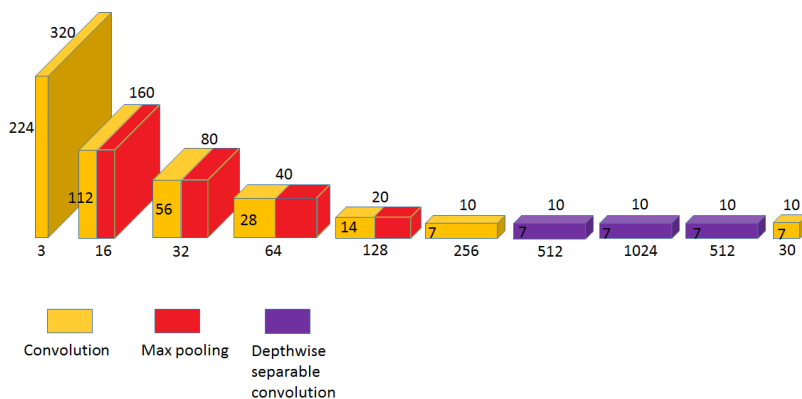


FIGURE 7. Architecture of the modified YOLO v2-tiny.

falls; and false negative (FN) is the number of falls that are mistaken for normal activities.

In the performance evaluation of YOLO and the overall system (YOLO and SVM), the PC and edge computing platform were used to evaluate and compare the performance of different YOLO models, respectively. The overall experimental process is detailed in Fig. 10.

#### IV. RESULTS

##### A. EXPERIMENTAL RESULTS FOR THE ORIGINAL AND MODIFIED YOLO MODELS

The key part of this system is the design of the AI model of the bounding human body. If an AI model with effective performance can be selected for use, the overall fall detection function is greatly improved. The AI model used for



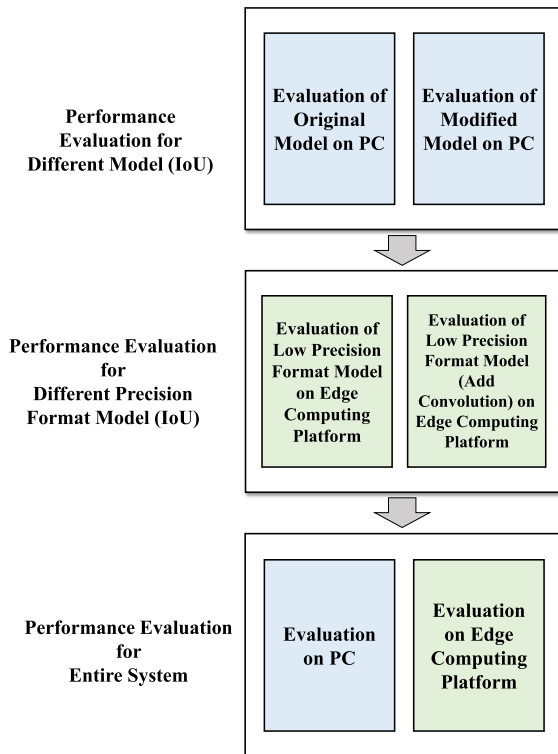


FIGURE 10. Overall experimental process for the performance evaluation of different AI models, stages, and platforms.

TABLE 2. Comparison between the modified Yolo v2-tiny and modified Yolo v3-tiny IoU.

Iteration	Modified YOLO v2-tiny (%)	Modified YOLO v3-tiny (%)
1	79.02	94.28
2	73.24	95.41
3	73.96	96.73
4	74.36	95.39
5	73.71	95.74
Average	74.86	95.51

YOLO v3-tiny with integer 8 precision format and a convolutional layer (Fig. 9) on the edge computing platform. The comparison of the experimental results is detailed in Table 3. The float 16 precision format model used the CPU on the edge computing platform to infer the neural network. The IoU was 94.6%, which is 0.9% lower than that of the computer version of the float 32 precision format model at 95.51%. The integer 8 precision format model used the KPU on the edge computing platform to infer the neural network; the resultant IoU of 91.2% was 4.3% lower than that of the float 32 precision format model. Although the IoU of the float 16 precision format model was higher than that of the integer 8 model, the FPS of the integer 8 model was 14.7 times that of the float 16 model. Following the addition of the convolutional layer to the integer 8 precision format model, the IoU increased to 94.5% and FPS decreased by 0.3%. Among the three models, the integer 8 precision format model

TABLE 3. Comparison of different precision format models on edge computing platform.

	Float 16 (CPU on Edge Computing Platform)	Int 8 (KPU on Edge Computing Platform)	Int 8 + Convolutional Layer (KPU on Edge Computing Platform)
Average IoU (%)	94.6	91.2	94.5
FPS	0.8	11.8	11.5
MB	3.3	1.6	3.1

is the smallest at approximately half the size of the other two models.

C. SYSTEM EVALUATION

The first step of the proposed system is human bounding. Table 3 presents the performances of bounding models. The final result shows that model on edge computing platform can reach 94.5% average IoU with 11.5 FPS.

The second step of the proposed system is SVM classification. Table 4 presents the performance of bounding model and SVM classifier. System spends less than 0.001 s on feature extraction and SVM classification. The system using YOLO v3-tiny achieved an accuracy of 92.5% on the PC, which is almost the same as the classification result using the ground truth. After part of the convolutional layer was replaced with a deep separated convolutional layer, the modified YOLO v3-tiny achieved an accuracy of 91.6%, a decrease of 0.9%. The modified ability of YOLO v3-tiny to detect objects with curved bodies was slightly reduced, but the classification was largely correct. When the modified YOLO v3-tiny was converted to the integer 8 precision format, the information of the neural network was lost, resulting in a decreased IoU. The addition of a convolutional layer can effectively improve the performance of the neural network, which forms a new model, YOLO-LW. In comparison with the system that used the modified YOLO v3-tiny, the system that used YOLO-LW exhibited a slight decrease in accuracy (0.5%).

The final step of the proposed system is fall event detection. A sliding window is used to detect fall event. When the falling state appears more than three times in a sliding window, the result is considered as a fall. Fig. 11 shows performance under different sizes of sliding windows. According to the result, system reaches the highest performance when size of sliding window is four. A sliding window takes four images to consider a fall event. Therefore, entire system spends 0.344 s to process a fall event.

V. DISCUSSION

In this study, a fall detection system that combined a camera and neuromorphic computing hardware was investigated. Although research demonstrated that the implementation of deep learning in automatic fall detection systems can enhance fall detection performance [18], the use of deep learning in embedded systems greatly increases the computing time. This study is the attempt to use neuromorphic computing

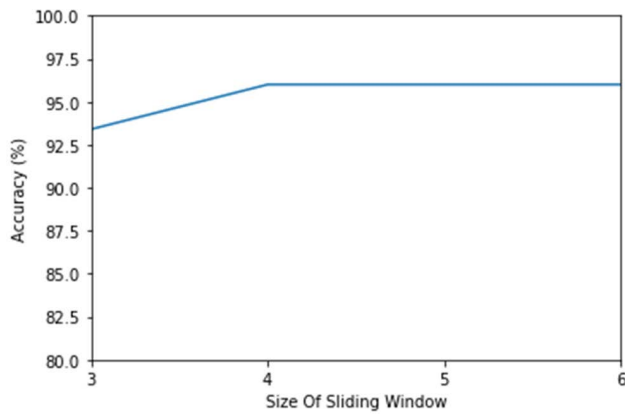


**TABLE 4. Comparison of SVM and different models.**

Platform	Acc. (%)	Prec. (%)	Rec. (%)	Spec. (%)	F1-score (%)	FPS
YOLO v3-tiny + SVM (on PC)	92.5	96.2	90.3	95.5	93.1	-
Modified YOLO v3-tiny + SVM (on PC)	91.6	95.8	89	94.9	92.2	-
YOLO-LW + SVM (on Edge Computing Platform)	91.1	95.5	88.5	94.5	91.8	11.5

**TABLE 5. Performance of whole system.**

Acc. (%)	Prec. (%)	Rec. (%)	Spec. (%)	F1-score (%)	Detection time (s)	Power consumption (W)
96	97.2	94.5	97.4	95.8	0.344	0.3



**FIGURE 11. Results under different sizes of sliding windows.**

hardware to replace the software for deep learning in embedded systems. We executed our self-developed YOLO-LW on neuromorphic computing hardware to maintain the running time of the neural network with the integer 8 precision format, without losing the region of interest (ROI) changes related to possible fall events.

To evaluate the fall detection system involving the ROI, various action photos collected from five indoor scenes were transmitted to five YOLO models with different neural layer structures, and their performances were compared through 5-fold cross-validation. From the experimental results of Table 1, at a resolution of  $320 \times 224$ , the YOLO v3-tiny outperformed the YOLO v2-tiny. This is attributable to the YOLO v3-tiny use of upsampling to obtain high-scale feature maps and retain the information of small targets. As described in Table 2, the modified YOLO v3-tiny exhibited greater accuracy than the modified YOLO v2-tiny but exhibited 2.65% less accuracy compared with that of the YOLO v3-tiny. The modified YOLO v3-tiny used a deep separation of convolutional layers to reduce the complexity of the model; thus, the accuracy was slightly reduced, but the model size was also reduced by one-fifth.

In the fall detection system, the bounding of the human body is a critical technology. Generally, an improved IoU increases the hit rate of human body area, and a reduction in IoU often occurs when the detection object is self-occluded during bending of the body, such as having the individual having their back to camera or curling their limbs. Additionally, in the process of porting the PC-side programs to the edge computing platform, some components may be abandoned, causing IoU decline. Some studies have indicated that a remapping of the neural network with high precision format to the neural network with low precision format can effectively reduce the model size and maintain the accuracy [23], [24]. The experimental results outlined in Table 3 demonstrated that the modified YOLO v3-tiny with float 16 format exhibited only a 0.91% decrease in IoU than the model before conversion, but the model size is half that of the original. However, the modified YOLO v3-tiny with float 16 precision format could not make inferences in the neuromorphic computing hardware and only reached 0.8 FPS with CPU inference. The modified YOLO v3-tiny with integer 8 precision format used neuromorphic computing hardware to infer and reached 11.8 FPS, but the IoU decreased by 4.3%. Despite the robustness of the modified YOLO v3-tiny with integer 8 precision format and its ability to detect all objects in different scenes and with different actions, the IoU decreased markedly. If the complexity of the model is increased through the addition of a convolutional layer, the IoU can reach 94.5%; the model size would be smaller than the modified YOLO v3-tiny with the float 16 precision format, and the speed only 0.3 FPS slower than the modified YOLO v3-tiny with integer 8 precision format.

The experimental results of Table 4 revealed that the fall detection system composed of YOLO v3-tiny and SVM and implemented on the PC side, with a GPU used for inference, achieved an accuracy rate of 92.5%. However, the high costs of the GPUs and desktop computers make it impossible for the fall detection system to be deployed flexibly in actual scenarios. Despite the PC acting as a server, the system still faces problems with network delays and processing large amounts of data. The system implemented on the PC and composed of the modified YOLO v3-tiny and SVM exhibited a decreased IoU, but the overall accuracy only decreased by 0.9%. However, after the modified YOLO v3-tiny was converted to the integer 8 precision format model, the accuracy of the overall system was considerably reduced. The system composed of YOLO-LW and SVM on the edge computing platform achieved an accuracy rate of 91.1% and a speed of 11.5 FPS. Among the various indicators, recall was the only one lower than 90%. False alarms mostly occurred when most of the skin area of the subject was occluded, causing the system to confuse the clothing of the subject with the background and partially deforming the bounding box.

Table 6 presents a comparison between our proposed method and those in other studies based on edge computing. Yu *et al.* [19] proposed a wearable fall detection system based on neuromorphic computing hardware. The Hopfield

**TABLE 6. Comparison of the proposed method and other fall detection systems.**

	Proposed Method	Yu <i>et al.</i> [19]	Yang <i>et al.</i> [30]	Mauldin <i>et al.</i> [31]	Alaoui <i>et al.</i> [32]	Chang <i>et al.</i> [33]
Algorithm	YOLO-LW + SVM	Hopfield neural network	CNN	RNN	PCA + SVM	OpenPose-light + SVM
Environment	SoC (RISC-V + KPU)	PSpice	FPGA	ARM	PC	SoC (ARM + GPU)
Sensor	Camera	IMU	Camera	Accelerometer	Camera	Camera
Processing time	0.344 s	Exceeds 5 s	0.413 s	1.28 s	-	4 s
Power consumption	0.3 W	-	2.5 W	-	Large	15 W
Performance	Accuracy: 96% Recall: 94.5% Specificity: 97.4% Precision: 97.2% F1-score: 95.8%	Accuracy: 88.9%	Accuracy: 92%	Accuracy: 70% Recall: 86% Precision: 64%	Accuracy: 98.5% Sensitivity: 97% Specificity: 100%	Accuracy: 98.1%

neural network was simulated using PSpice as a circuit of the neuromorphic computing hardware. The system analyzed the IMU data to determine falls with an accuracy of 88.9%, though the author performed a circuit simulation rather than constructing the wearable device. Yang *et al.* [30] used field programmable gate array (FPGA)-based ZYNQ-7020 hardware to implement a CNN model with an accuracy rate of 92%, but the power consumption of 2.5 W is too costly for a fall monitoring system that must operate for a long time. The detection time was also 0.43 s, with an FPS of only 2.42. Mauldin *et al.* [31] employed three-layer open system architecture to transmit the sensor data from a smart watch to a smartphone for edge computing. They implemented a recurrent neural network on a smartphone based on an ARM processor, but an accuracy of only 70% was achieved. The method of Alaoui *et al.* [32] is to first calculate the key points of human skeleton, and then use principal component analysis (PCA) and SVM to detect whether someone has fallen in the image data. The accuracy of the experimental results of their algorithm is 98.5%. The overall performance is good, but the entire study is based on an analysis of a ready-made dataset, and it is impossible to confirm the performance in the real environment. The system will transmit the video to the server, there will be the problem of personal privacy leakage. Chang *et al.* [33] constructed OpenPose-light and SVM algorithms on an edge computing platform (Jetson TX2, NVIDIA Corp., Santa Clara, CA, USA) to detect falls for elderly people with an accuracy of 98.1%. Its overall performance is good, and the use of edge computing can avoid personal privacy issues. But its processing time is a bit of long and power consumption is a bit of high.

Overall, the method proposed in this study was more accurate than the aforementioned methods. In terms of use, we used a camera as the input device, which is more convenient than a wearable fall detection system using IMU. Regarding the same vision-based solution, compared with the 2.5 W and 15 W power consumption reported in [30] and [33] respectively, the architecture proposed in this study required 0.3 W, providing a low-cost, low-computing, feasible resource allocation solution. In terms of detection speed,

this system reached 11.5 FPS, which provided effective real-time performance. Additionally, the FPGA debugging process is difficult and extends the development time.

## VI. CONCLUSION

In this study, a fall detection system with neuromorphic computing hardware for AI-based edge computing was proposed. The images of individuals were captured through the camera and transmitted to the neural network model on the edge computing platform. After detection of the object characteristics, the SVM was used for classification, and the detection result was communicated to the manager via Wi-Fi. This study successfully deployed an improved neural network model YOLO-LW on the edge computing platform. YOLO-LW uses a deep separated convolutional layer to improve computational efficiency, differing from the model with float 32 precision format on the computer side. YOLO-LW is converted to integer 8 precision format to increase FPS and reduce model size, with an additional convolutional layer added to maintain the accuracy of the model. In the experiment, we collected normal and falling photos of people of all ages under five different indoor backgrounds through the camera of the platform and fed these images to the model for training and verification to validate the robustness of the proposed method. After experimental evaluation, an average IoU of 94.5% was obtained on the edge computing platform; the accuracy of the overall system reached 96%, and the FPS reached 11.5. It exhibited excellent real-time performance and a power consumption of only 0.3 W. Power consumption is a crucial factor for fall monitoring systems that must be operated for a long time. All data were calculated on the edge computing platform, thereby protecting the privacy of users. Despite occlusion problems, the proposed neural network has good generalizability. At a distance of 2–3.5 m, the object can still be captured even if one-third is occluded, which presents a potential edge computing solution. The proposed framework is client server-based and single-tier architecture [34], which assist with cost savings and safety. The proposed framework also satisfies network bandwidth saving and real-time data processing [35]. However, complex occlusion situations and

variety of light may affect performance of fall detection system.

In future work, the fall detection system can incorporate the use of different sensors, such as a thermal camera installation to monitor the activities of older adults in lightly lit environments and at night or a fisheye lens installation to expand the detection range. In addition, different shooting angles and complex occlusion situations can be further evaluated.

## REFERENCES

- [1] S. J. Naja, M. M. Makhlof, and M. Chehab, "An ageing world of the 21st century: A literature review," *Int. J. Community Med. Public*, vol. 4, no. 12, pp. 4364–4369, 2017.
- [2] C. S. Florence, G. Bergen, A. Atherly, E. Burns, J. Stevens, and C. Drake, "Medical costs of fatal and nonfatal falls in older adults," *Ann. Long-Term Care*, vol. 66, no. 4, pp. 693–698, Jul. 2018.
- [3] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 2, pp. 290–300, Mar. 2011.
- [4] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 314–323, Jan. 2019.
- [5] E. Akagunduz, M. Aslan, A. Sengu, H. Wang, and M. C. Ince, "Silhouette orientation volumes for efficient fall detection in depth videos," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 756–763, May 2017.
- [6] Z.-P. Bian, J. Hou, L.-P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 2, pp. 430–439, Mar. 2015.
- [7] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "An integrated vision-based approach for efficient human fall detection in a home environment," *IEEE Access*, vol. 7, pp. 114966–114974, 2019.
- [8] B.-H. Wang, J. Yu, K. Wang, X.-Y. Bao, and K.-M. Mao, "Fall detection based on dual-channel feature integration," *IEEE Access*, vol. 8, pp. 103443–103453, 2020.
- [9] A. Lotfi, S. Albawendi, H. Powell, K. Appiah, and C. Langensiepen, "Supporting independent living for older adults: Employing a visual based fall detection through analyzing the motion and shape of the human body," *IEEE Access*, vol. 6, pp. 70272–70282, 2018.
- [10] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, May 2018.
- [11] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6722–6747, Aug. 2020.
- [12] K. Ozcan and S. Velipasalar, "Wearable camera and accelerometer-based fall detection on portable devices," *IEEE Embed. Syst. Lett.*, vol. 8, no. 1, pp. 6–9, Mar. 2015.
- [13] A. Shahzad and K. Kim, "FallDroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Trans. Ind. Inform.*, vol. 15, no. 1, pp. 35–44, Jan. 2018.
- [14] 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. 2019.
- [15] G. Shi, C. S. Chan, W. J. Li, K.-S. Leung, Y. Zou, and Y. Jin, "Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier," *IEEE Sensors J.*, vol. 9, no. 5, pp. 495–503, May 2009.
- [16] K. Ozcan, S. Velipasalar, and P. K. Varshney, "Autonomous fall detection with wearable cameras by using relative entropy distance measure," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 1, pp. 31–39, Feb. 2017.
- [17] J. H. Kim, "Embedded real-time fall detection using deep learning for elderly care," in *Proc. 31st Int. Conf. Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 2110–2118.
- [18] A. El Kaid, K. Baina, and J. Baina, "Reduce false positive alerts for elderly person fall video-detection algorithm by convolutional neural network model," *Proc. Comput. Sci.*, vol. 148, pp. 2–11, May 2019.
- [19] Z. Yu, A. Zahid, S. Ansari, H. Abbas, and A. Abdulghani, "Hardware-based Hopfield neuromorphic computing for fall detection," *Sensors*, vol. 20, no. 24, 2020, Art. no. 7226.
- [20] 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.*, pp. 1–13, Mar. 2021, doi: 10.1007/s12652-021-03157-1.
- [21] S. Carata, R. Mihaescu, E. Barnovicu, 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. (ICCP)*, Sep. 2019, pp. 517–523.
- [22] V. Leon, S. Mouselinos, K. Koliogeorgi, S. Xydis, D. Soudris, and K. Pekmestzi, "A tensorflow extension framework for optimized generation of hardware CNN inference engines," *Technology*, vol. 8, no. 1, Jan. 2020, Art. no. 6.
- [23] 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, 2021, Art. no. 2984.
- [24] 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.
- [25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.
- [26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [27] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.
- [28] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1251–1258.
- [30] X. Yang, S. Tang, T. guo, K. Huang, and J. Xu, "Design of indoor fall detection system for the elderly based on ZYNQ," in *Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, Chongqing, China, Dec. 2020, pp. 1174–1178.
- [31] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. H. Ngu, and C. C. Rivera, "SmartFall: A Smartwatch-based fall detection system using deep learning," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3363.
- [32] A. Y. Alaoui, S. E. Fkhi, 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.
- [33] 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.
- [34] R. M. A. Haseeb-Ur-Rehman, M. Liaqat, A. H. M. Aman, S. H. A. Hamid, R. L. Ali, J. Shuja, and M. K. Khan, "Sensor cloud frameworks: State-of-the-art, taxonomy, and research issues," *IEEE Sensors J.*, vol. 21, no. 20, pp. 22347–22370, Oct. 2021.
- [35] E. Ahmed, A. Ahmed, I. Yaqoob, and J. Shuja, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, Nov. 2017.



**BOR-SHING LIN** (Senior Member, IEEE) received the B.S. degree in electrical engineering from National Cheng Kung University, Taiwan, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taiwan, in 1999 and 2006, respectively. Since 2009, he has been an Assistant Professor with the Department of Computer Science and Information Engineering, National Taipei University, Taiwan. He is currently a Professor with the Department of

Computer Science and Information Engineering, National Taipei University. He is also the Director of the Computer and Information Center, National Taipei University. His research interests include smart medicine, embedded systems, wearable systems, biomedical signal processing, biomedical image processing, and portable biomedical electronic system design.



**TIKU YU** received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2001 and 2003, respectively, and the Ph.D. degree in electrical engineering from the University of California at San Diego, La Jolla, CA, USA, in 2009. From 2009 to 2010, he was a Post-doctoral Researcher with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. From 2010 to 2013, he was a Technical Staff with Mediatek USA in San Jose, California. Since 2013, he has been with the Department of Communication Engineering, National Taipei University, Taiwan, where he is currently an Associate Professor. His research interests include design and development of RFIC, MMIC, and microwave circuits.



**HUNG-KAI HSU** is currently pursuing the M.S. degree with the Department of Computer Science and Information Engineering, National Taipei University, Sanxia, New Taipei City, Taiwan. His research interests include embedded systems and artificial neural network applied to biomedical engineering.



**CHIH-WEI PENG** received the B.S. degree in physical therapy from Chang Gung University, Taoyuan, Taiwan, in 2000, and the M.S. and Ph.D. degrees in biomedical engineering from National Cheng Kung University, Tainan, Taiwan, in 2002 and 2007, respectively. He is currently a Full Professor and the Chairperson with the School of Biomedical Engineering, Taipei Medical University. His research interests include neural engineering, biomaterial, bio-signal process,

biosensor, rehabilitation engineering, neural prostheses, urological electrophysiology, and functional electrical stimulation.



**I-JUNG LEE** received the B.S. degree in computer science and information engineering from National Taipei University, Sanxia, New Taipei City, Taiwan, in 2014, where she is currently pursuing the Ph.D. degree with the College of Electrical Engineering and Computer Science. Her research interests include embedded systems and wearable devices applied to biomedical engineering.



**CHUEH-HO LIN** received the B.S. degree from the Department of Physical Therapy, Hung Kung University, Taiwan, in 2005, and the M.S. and Ph.D. degrees from the Department of Physical Therapy and Assistive Technology, National Yang-Ming University, Taiwan, in 2007 and 2014, respectively. Since 2015, he has been an Assistant Professor with the Master Program in Long-Term Care, Taipei Medical University, Taiwan. He is currently the Director and an Associate Professor

with the Master Program in Long-Term Care, Taipei Medical University. His research interests include developing the novel assistive devices with multiple sensors to investigate age-related changes in muscle strength generation, bilateral limbs coordination, motor control, and brain-behavior relationships during specific task in both non-disabled individuals (health young and elderly adults) and individuals with neurologic disorders.



**ZHAO ZHANG** (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Hefei University of Technology, China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree with of the College of Electrical Engineering and Computer Science, National Taipei University, Taiwan. After graduating, he was the Principal Engineer at EDAN Instruments, from 2008 to 2009. Since 2011, he has been a

Lecturer with the College of Mechanical and Electrical Engineering, Wuyi University, China. His research interests include smart medicine, embedded systems, wearable systems, and portable biomedical electronic system design.

...