IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 27, 2019, accepted October 5, 2019, date of publication October 15, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947518

Implementation of Fall Detection System Based on 3D Skeleton for Deep Learning Technique

TSUNG-HAN TSAI, (Member, IEEE), AND CHIN-WEI HSU
Department of Electrical Engineering, National Central University, Taoyuan City 32001, Taiwan

Corresponding author: Tsung-Han Tsai (han@ee.ncu.edu.tw)

This work was supported in part by the DSP Lab Research Team, National Central University, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-008-078-MY3.

ABSTRACT In recent years, the fall detection system has become an important topic in the homecare system. Compared with the traditional fall detection algorithm, the method used by neural network is more robust and has higher accuracy. However neural network consumes a large amount of energy due to a huge number of computations, and needs more memory to store parameters as compared to traditional algorithms. In this paper, we propose a fall detection system in combination of the traditional algorithm with the neural network. First, we propose a skeleton information extraction algorithm, which transforms depth information into skeleton information and extracts the important joints related to fall activity. Also we have modified the skeleton-based method with seven highlight feature points. Second, we propose a highly robust deep convolution neural network architecture, which uses a pruning method to reduce parameters and calculations in the network. The low number of parameters and calculations makes the system suitable for the implementation on an embedded system. The experiment results show the high accuracy and robustness on the popular benchmark dataset NTU RGB+D. The proposed system has been implemented on NVIDIA Jetson Tx2 platform with real-time processing.

INDEX TERMS 3D skeleton, action recognition, deep learning, fall detection, embedded system, image processing.

I. INTRODUCTION

With the population-aging index rising, the social welfare of the elderly people needs more attention, and safety care is becoming more and more important. Falling is one of the major causes of death among elderly people. According to statistics, more than 20 percent of elderly people fall every year. Among many reasons for fall, some include falling due to the surrounding environment, or the body loses its balance when the elder person is moving, or because of their personal medical history. However, whatever the cause of the fall is, it will cause physical and mental injury to the elderly people.

In many cases, the consequences of a fall in the elder people are unimaginable, such as fractures, head injuries, etc. Even if falling does not cost some serious physical injury, it can cause a so-called post-fall syndrome [1]. Because of the fear of falling down, elder people begin to limit their movement, resulting in a gradual loss of physical activity. The study has shown that getting up quickly after a fall can reduce the risk of death by 80% [2]. If the elderly person lives alone and in case of falling down, it will cause irreparable damage if someone nearby is not available to help the fallen person to get up. Therefore, to reduce the injury chances, fall detection can be performed as soon as possible when an elderly person falls down.

As for the families of the elderly people who live alone, their families tend to think about the safety of their elders and they want to send the elderly people to a nursing home so their better care can be taken. But many elderly people do not want to leave their home to go into a strange environment. By taking above-mentioned points into the consideration, for those old people who live alone and whose families are often out of their homes, a smart home system with such a 24-hour fall detection and early warning system can ensure the safety of the elderly people.

There are many ways to detect falls, which are divided into two sensors categories i.e., wearable sensors and environmental sensors. A wearable device detects falling through the calculation of body position change, acceleration change, etc., and issues a warning. The problem in this type of sensor is the occurrence of false alarm alerts. While using this device, the false alarm may trigger by an everyday activity such as

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

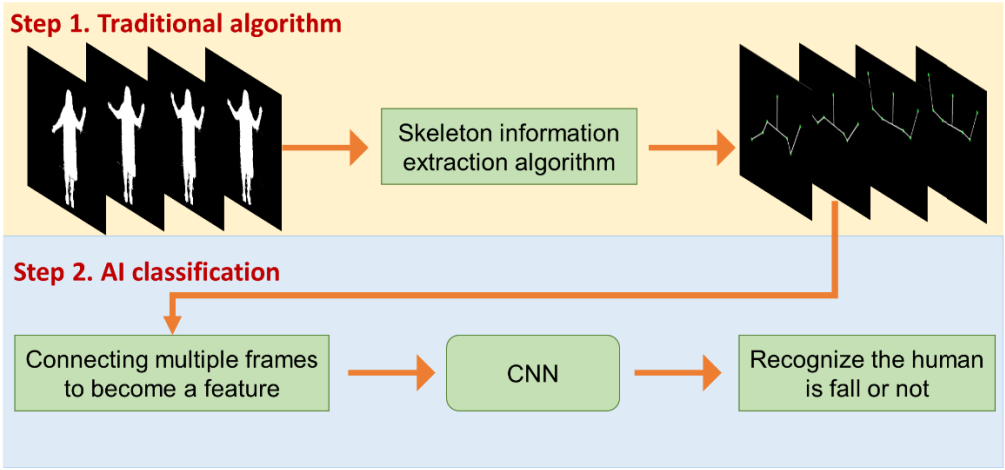

FIGURE 1. System overview.

squatting down or falling into bed quickly. The environmental sensor detects falls in a non-invasive manner by using pressure, vibration, and visual information. Compared with the wearable device, the environmental sensor is free from the inconvenience of wearing the device, yet the false alarm may occur.

The neural network is one of the machine learning algorithms, which are inspired by biological neural networks. There are multiple artificial neurons to perform the calculations and a learning function to train the network through a large amount of information. Neural networks are being used to solve various classification problems, including fall detection. Compared with the traditional threshold-based fall detection method, the fall detection method based on the artificial neural network can achieve higher precision for fall detection [3]. Artificial neural network can effectively improve the accuracy and reliability of visual-based fall detection and warning systems.

Although the accuracy of the fall detection method is high, yet it results in a large number of convolution operations which are hindrance to implement the system in an embedded system. Most of the fall detection algorithms have been implemented using computers with GPU accelerator [4]–[8]. So it affects the mobility factor of the system and system cannot be implemented at all the places. In order to use the fall detection system widely in society, the proposed system can perform well on embedded systems by reducing the number of parameters and bringing precision in an acceptable range.

In this paper, we propose a video-based fall detection system. In order to achieve real-time processing, we combined the traditional algorithms and AI classification technique. The traditional algorithms of preprocessing are used to extraction the most relevant seven skeleton features as the network input by the depth information from the camera. AI classification is used with pruning method to reduce the parameters of the neural network and maintain the higher accuracy for the fall detection system.

Our goal is to implement the proposed algorithm on an embedded system, NVIDIA Jetson TX2, for real-time demonstration. To capture the moving activity, we use a depth image camera, Microsoft's Kinect V2 module, to acquire the depth information. Since the current Kinect V2 SDK cannot support ARM 64 processor, we use Openni2 framework to read 640 × 480 depth image form Kinect V2 module. Then the 3D skeleton extraction algorithm is applied and combined with deep neural network method to determine whether the people fall or not.

An overview of the proposed system is shown in Fig. 1. With respect to the overview, the architecture of the article is organized as follows. Section 2 discusses the related work. Section 3 explains the details about the extraction of skeleton information. Section 4 describes the deep neural network architecture. Section 5 presents the experimental results. And section 6 presents the conclusions.

II. RELATED WORK

The identification of different human activities has been extensively studied over the past few years. Through the rapid development of science and technology, more and more focus is being put on social welfare to improve people's life. The protection of elderly people living at home from different dangerous situation such as falls is quite important.

A. WEARABLE SENSOR AND ENVIRONMENTAL SENSOR

In the area of fall detection, it has been mainly divided into the wearable sensor and environmental sensor categories. Table 1 shows some the related works. A wearable device uses a sensor to determine whether a fall has occurred. Some examples of such sensors are tri-axial accelerometer [9], gyroscope [10], [11], etc. Some wearable devices include three-axis accelerometer, three-axis gyroscope and three-axis magnetometer [12]. These sensors, which has been designed based on the physical devices, detect the change in people's position. However, it may be costly and inconvenient because users have to wear such sensors all the time.

The environmental sensor is used to analyze and process the video stream using the algorithm to determine whether the fall has occurred or not. Environmental sensors can be divided

TABLE 1. Technology on fall detection.

	Reference	Technology	Cost	Risk on privacy	Size	Accuracy	Recognition site
Wearable sensor	[9]-[12]	tri-axial accelerometer, Gyro	High	Low	Medium	Low	Sensor
Environmental sensor	[13]-[15]	Surveillance based system	Low	High	Medium	Low	Sensor
	Proposed	Vision with interaction	High	Low	Large	High	Sensor/ Server

into a surveillance-based system and vision with interaction. The surveillance-based system can automatically monitor the home environment to perform fall detection. [13], [14] propose the algorithm that combines multiple camera surveillance systems to detect falls. In [15] it proposes a multi-object tracking algorithm for fall detection.

Vision with interaction usually contains multiple visual sensors for multi-sensor use, such as the IR image sensor, stereo image sensor, single image sensor, etc. A two-person activity recognition which used the skeleton extraction to gain the features from a depth camera is proposed [16]. The composition of Kinect is composed of a depth sensor, RGB camera and microphone array. It can be used to check depth information, schema skeleton, language input, etc. This shows that Kinect can be used in applications related to the visual sensors, such as face recognition [17], human action recognition [18], hand gesture recognition [19] etc. By Microsoft Kinect's depth information, [20] propose a new skeleton-based approach to describe the spatiotemporal aspects of a human activity sequence as a monitoring system for the elderly.

B. TRADITIONAL ALGORITHMS AND AI

The operation flow of the traditional fall detection has been described in [21], [22]. There are many methods of foreground segmentation, such as optical flow method [23], [24], frame difference [25], and background subtraction [26], [27]. Optical flow method is very sensitive to environmental noise. As a result, it is difficult to take into account as the accuracy issue. Also the slow processing makes it had to implement in real-time processing. The frame difference cannot cut out the object completely, so it becomes vulnerable to produce an error during the detection process. Background subtraction can also make it difficult to accurately cut the object because of changes in environmental noise. In all the above-mentioned methods, the object is cut out first and the information is used to detect the fall. This may lead to the lower precision of the traditional fall detection algorithms.

In the identification part, the neural network can achieve higher accuracy compared with traditional algorithm [28], [29], because the neural network can obtain more and more features by the training. The convolutional neural network is the basis of many image identification models, which is usually made up of the convolutional layer, pooling layer, etc. [30] Depth features are extracted from multiple convolutions and pooling layer reduces parameters and calculations. In order to avoid overfitting, the dropout is included to improve the results.

In [31] which combined the background subtraction algorithm and the deep learning technique. The network module used the silhouette normalization as the input. It discussed on different phases at the falling time to decision the falling case carefully. The system could be manufactured at a low cost. However, the feature of the silhouette is not superior for the falling detection, because of changes in environmental noise by the background subtraction algorithm and some information is not essential. In order to solve this problem, we propose a method which can obtain the important features of the falling situation as the input of the network to achieve higher accuracy of the system.

The video-based fall detection method proposed in [32] used neural network along with a large number of datasets for training. The skeleton information is extracted by the OpenPose [33] which is able to run at around 22 FPS in the machine with the GPU, and the Deep neural networks and SVM used for classification. However, the skeleton information is extracted by the VGG16 and multi-stage CNN which have higher than 150M parameters. Also it needs the high-performance GPU to store the parameters and accelerate the neural network processing. Consequently, it is not suitable in the embedded system. We propose the skeleton extraction method without the GPU, and apply some pruning technique to reduce the parameter of the network to fit the embedded system. The system overall is shown in Fig. 2. It is composed with skeleton information extraction by traditional algorithm, and action recognition made by neural network algorithm. The network structure is shown in Fig. 3.

III. SKELETON INFORMATION EXTRACTION ALGORITHM

The proposed skeleton extraction system is divided into four steps as shown in Fig. 2. The four steps are named as foreground segmentation, labeling, thinning and searching. Because the deep neural network is used to classify the falling event, the accurate absolute position of every node is not

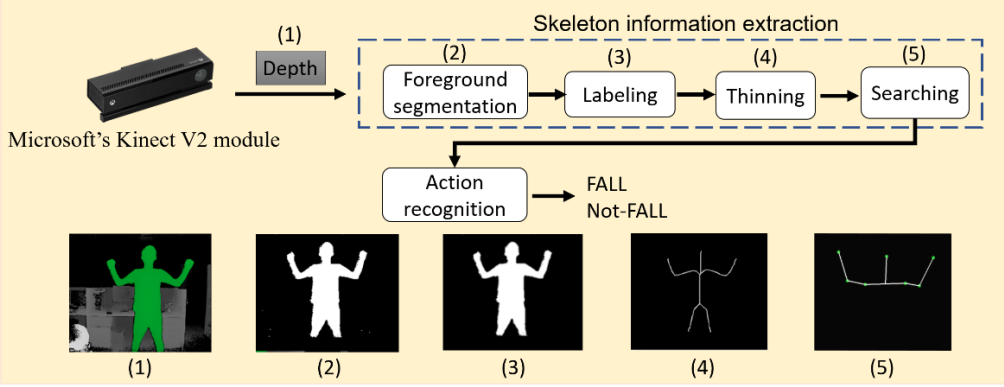

FIGURE 2. Proposed overall fall detection system.

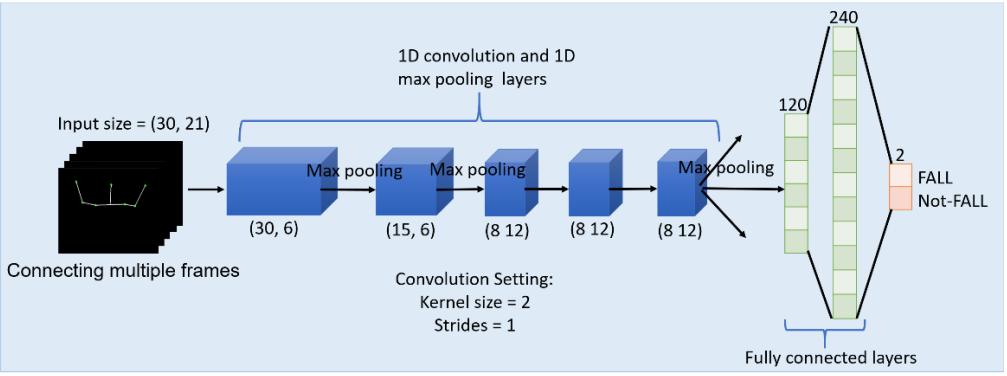

FIGURE 3. The neural network structure.

needed. The relative position of important nodes can certainly achieve high-performance results. By not getting the absolute position points, the computation time of processing and the information can be reduced which is helpful to implement the model on an embedded system.

A. FOREGROUND SEGMENTATION

Foreground segmentation has quite complex computations due to the use of complex algorithms during object detection. Gaussian mixture model (GMM) is one of the kernels to perform foreground segmentation [34], [35].

In GMM, if the object does not move for a long time, it will be considered as background. It will lead to the failure to recognize the fall event correctly in some case. Other complex algorithms may be used to resolve this problem. However, the high complexity issue creates hindrance to implement the design on an embedded system.

In the proposed method, GMM is not used for foreground detection. We use depth information which is captured by Microsoft's Kinect V2 module to obtain foreground. Then we scan through the depth information of each pixel and get the closest point which is most likely the position of the person and then 600 mm region is marked as foreground in front of the camera. The acquired foreground is plotted as a green

FIGURE 4. The depth image of the foreground. (a) Depth image (b) after resizing and binary.

color, as shown in Fig. 4 (a). We resize this foreground result into a 320 × 240 binary image, as in Fig. 4 (b), to reduce the error after the skeleton point has been obtained. A reduced size skeleton is also helpful to process the information in a deep learning algorithm. Since a lot of noise will be generated during the labeling process, the accuracy of this foreground detection method may not be better than [34] and [35] at this stage. So, the noise has been removed in the second and third step.

B. LABELING

Based on foreground segmentation, we can obtain the interested region of the binary image. Although some noise occurs

in the image, object labeling is applied to label the foreground object as the same tag as shown in Fig. 5. It cannot only remove the rest of the noise, but it can also calculate the coordinates, length, and width of the foreground object. We use the information of area size to determine whether the object is human or not. Thus it can solve the problem when the camera is closer to the non-human object.

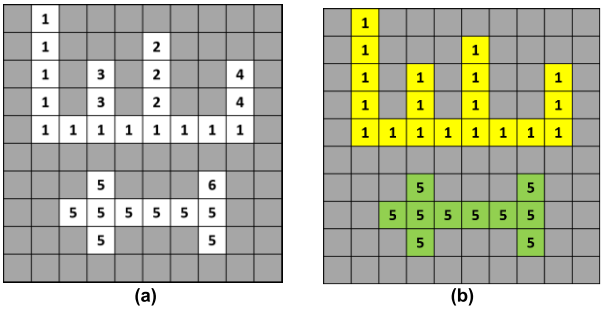

FIGURE 5. Labeling diagram of the (a) before labeling (b) after labeling.

The traditional labeling algorithm is very slow and requires much memory to store different parameters. While considering a real-time implementation, we use our previous work of [36], which is a combination of running length coding (RLC) and object labeling to perform the noise removal. This technique can reduce the computation time yet it maintains the performance of the system.

C. THINNING

There are many methods for thinning in the binary image [37], [38]. In this paper, we use Zhang Suen's rapid thinning method [37]. Compared with other thinning methods, it can achieve rapid thinning with high processing speed so it is able to process data in real-time. Based on the object block obtained by labeling algorithm, the thinning is performed as shown in Fig. 6 (a). Since the result still contains too much noise, the dilation and erosion algorithm are executed to perform noise removal [39]. The image of thinning result after removing the extra noise is shown in Fig. 6(b). The final result is shown as a human doll.

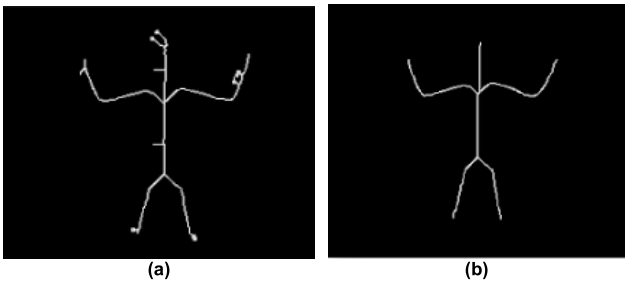

FIGURE 6. The result of the thinning algorithm. (a) Before noise removal (b) after noise removal.

After the thinning, the skeleton information becomes clear which reduces the probability of miscalculation when the searching algorithm is performed later. Eventually, it improves the accuracy of the skeleton nodes.

D. SEARCHING

Fig. 7 shows a 3D skeleton which has been achieved using the proposed algorithm on Kinect software development kit V1, and Kinect software development kit V2. This complete set of joint format is provided by Kinect v1 SDK. Overall 20 points have been marked for a human body as the joints which have been shown in orange color in Fig. 7(a). In Fig. 7 (b), 25 points are marked as joints which have been achieved using the upgraded Kinect v2 SDK. The upgraded Kinect provides five extra points as compared to the previous version.

The skeleton nodes obtained by our searching algorithm are different from the above two SDK. We just use 7 joint points marked as the blue color shown in Fig. 7(c). The reason is that when a falling event will occur, the joints above the waist will have enormous change. This reduced joint-point set can reduce the training parameters as well as training time of neural network while performing decision about the fall event. First, we calculate the center of the body as the coordinate (x, y). In the second step, the blue points are calculated. The description of these seven blue points is provided in Table 2.

IV. THE NEURAL NETWORK

In this part, we present a newly developed neural network which uses low parameters and computations for fall detection through the training of action recognition datasets.

A. NTU RGB+D DATASET

The NTU RGB+D dataset [40] is the largest action recognition dataset. This dataset has been obtained using Kinect software development kit V2 and it includes 56880 action samples, containing 60 actions performed by 40 subjects, and the sensor was used to capture this data from the three angels: −45°, 0°, 45°. Each video contains RGB, depth, skeleton, and infrared information. Here we only need to determine whether the person is fall or not. Thus, we use the feature number 4, 5, 6, 8, 9, 10, 12{HEAD, SHOULDER_LEFT, ELBOW_LEFT, HAND_LEFT, SHOULDER_RIGHT, ELBOW_RIGHT, HAND_RIGHT}. These seven important nodes will be the input features of the neural network.

As a result, the complexity of the input and output dimension is reduced from the original 25 nodes to the current 7 nodes format. Additionally, there are totally 60 categories in NTU RGB+D database. Here, we just use the fall and non-fall categories for training purposes. Fig. 8 shows original samples from the NTU RGB+D dataset and our modified version's extracted joints results.

B. THE NETWORK ARCHITECTURE

The convolutional neural network currently converts the sequence of skeleton information into the corresponding actions [40]–[42]. It has high accuracy while doing action

TABLE 2. Description of the joint calculation.

Center	x	y
HEAD	x	Top of the Skelton
HAND_LEFT	The most left point of the Skelton	The most left point of the Skelton
HAND_RIGHT	The rightest point of the Skelton	The rightest point of the Skelton
SHOULDER_LEFT	Search for a first point whose pixel values is 255 from left to right, and from top to bottom, followed by following two conditions: 1. x is located between x of the head and the point 30 px left of the head. 2. y-axis is below 30px of y of the head	
SHOULDER_RIGHT	Search for the first point followed by two conditions below: 1. x is located between x of the head and the point 30 px right of the head. 2. y-axis is below 30px of y of the head	
ELBOW_LEFT	This point is the midpoint of the SHOULDER_LEFT and the HAND_LEFT.	
ELBOW_RIGHT	This point is the midpoint of the SHOULDER_RIGHT and the HAND_RIGHT.	

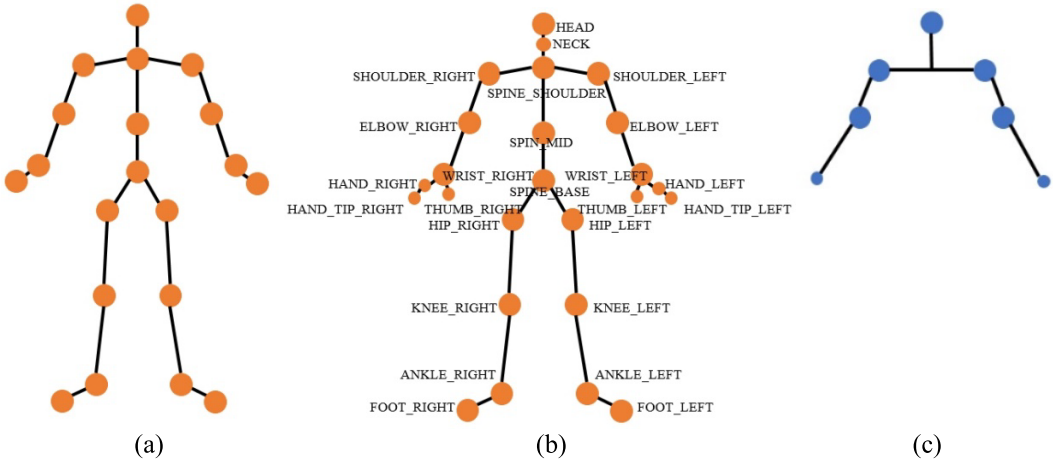

FIGURE 7. Compare skeleton information provided by different source. (a) Kinect SDK v1 (b) Kinect SDK v2 (c) Our skeleton extraction results.

TABLE 3. MyNet2D architecture.

	Type	Kernel Size / Stride	Output Size	Parameter	Ops
MyNet2D	Conv2D	5x5 / 1	30x21x12	300	189K
	Max Pool	3x3 / 2	15x11x12		
	Conv2D	3x3 / 1	15x11x12	1.3K	214K
	Max Pool	3x3 / 2	8x6x12		
	Conv2D	3x3 / 1	8x6x24	2.6K	125K
	Conv2D	3x3 / 1	8x6x24	5K	250K
	Conv2D	3x3 / 1	8x6x36	7.7K	375K
	Max Pool	3x3 / 2	4x3x36		
	FC		2048	884K	884K
	FC		2048	4M	4M
	Softmax		2		
Total				4.9M	6M

recognition. Here we only need to determine whether the fall event happened or not. Thus we only focus on fall detection.

In order to transfer the model into the embedded system, we first refer to the AlexNet [43] which has 5 convolution layers and 3 fully connected layers to construct the model. To implement the model on embedded board, model parameters are reduced to use in less resources of the board. Based on this architecture, we propose a network called MyNet2D as shown in Table 3, its input has 7 nodes and each node contains information about x, y, z coordinate so it will result in 21 information points for all nodes in the single frame. 30 frames are concatenated as a sequence which is used as the input, where the input size is (30,21). Here 30 represents the number of frames and 21 represents the information points of nodes. The number of parameters and calculations for this network are 4.9M and 6M respectively.

The skeleton information is the analog data, and there is no significant correlation between the horizontal axis and the vertical axis. Therefore, 2D convolution does not help much in this type of input data, and it results in a lot of unnecessary calculations and parameters. We further reduce the number of parameters through the reduction of the kernel size, filter

number, and input size. By using Conv1D, a simpler network MyNet1D-D has been designed which uses one-dimension to perform convolution to reduce parameters and perform calculations more effectively. By using this proposed architecture, the same accuracy can be achieved the model made up of 2D convolution.

Based on the optimizations, the proposed model is able to reduce different parameters as shown in Table 4. Compared with MyNet2D, the model architecture reduced the parameters from 4.9M to 35K and the number of calculations decreased to only 48K. Moreover, the experiment results show that our modified network has almost the same performance with AlexNet, MyNet2D. The modified network with a low number of parameters and a low number of calculations is more suitable for embedded systems.

TABLE 4. MyNet1D-D architecture.

	Type	Kernel Size / Stride	Output Size	Parameter	Ops
MyNet1D-D	Conv1D	2 / 1	30x6	252	7.5K
	Max Pool	2 / 2	15x6		
	Conv1D	2 / 1	15x6	72	1K
	Max Pool	2 / 2	8x6		
	Conv1D	2 / 1	8x12	144	1K
	Conv1D	2 / 1	8x12	288	2K
	Conv1D	2 / 1	8x12	288	2K
	Max Pool	2 / 2	4x12		
	Dropout		4x12		
	FC		120	5.8K	5.8K
	FC		240	29K	29K
	Dropout		240		
	SoftMax		2		
Total				35K	48K

Since the falling event videos are not much in the open dataset, it will cause the overfitting problem. In order to avoid the overfitting issue, we add dropout layer [44] in our network. We have used the exponential decay for the attenuation in the learning rate.

V. EXPERIMENT RESULTS

For training and testing purposes, we used NTU RGB+D dataset. We trained on an Intel Core i7, 3.60 GHz processor with 16GB of RAM and NVIDIA GTX1080Ti on a Linux system.

A. EVALUATION AND DISCUSSION

We first test the proposed neural network to find the accuracy rate for the fall detection event. Through the benchmark of NTU RGB+D dataset, all falling videos of skeleton information are treated as the positive samples. The other actions of skeleton information are randomly picked as the negative samples. Totally 946 videos belong to the positive samples, and 1419 videos are negative samples and 357 videos are testing sample. By keeping the negative samples higher than the positive samples, it can prevent the network from overfitting problem.

For the proposed network, we use 70 percent of the samples as training data, and 30 percent as the testing data. We use training data to train on the AlexNet, MyNet1D, and MyNet2D which do not have the dropout layers. The proposed network is trained by using gradient descent optimizer. Weights are initialized randomly and the learning rate of 0.01 is used. To attenuate the learning rate, the value is using exponential decay of 0.96 after every 2000 iterations. Iteration number is set as 50000 and the dropout rate is set as 0.5.

The accuracy rate is calculated using the Confusion Matrix as shown in Table 5. It determines the accuracy of the neural network. The TP, FN, FP, and TN represent the number of true positives, false negatives, false positives and true negatives, respectively. Accuracy is defined as the ratio between the number of correct classification samples divided by the total number of samples, which is represented as follows:

$$\mathrm{A} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

True positive rate or sensitivity is defined as recall, which can be defined as (2)

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{2}$$

While the precision of the network can be calculated using equation (3), which is as follows;

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{3}$$

The experiment results are shown in Table 6. After optimization, the result of MyNet1D-D has almost the same as the un-optimized models, but the number of parameters and computations, as compared to AlexNet and MyNet2D, are reduced significantly.

TABLE 5. Confusion matrix.

Actual \ Predicrted	Negative	Positive
True	True Negative	False Positive
False	False Negative	True Positive

TABLE 6. Performance comparison on accuracy, recall rate, precision on NTU RGB+D dataset.

	Accuracy	Recall rate	Precision	Ops	Parameter
AlexNet	98.9%	98.0%	99.3%	2.5G	60M
MyNet2D	98.8%	97.7%	99.2%	6M	4.9M
MyNet2D-D	99.4%	98.5%	99.9%	6M	4.9M
MyNet1D	98.8%	98.7%	98.7%	48K	35K
MyNet1D-D	99.2%	98.9%	99.1%	48K	35K

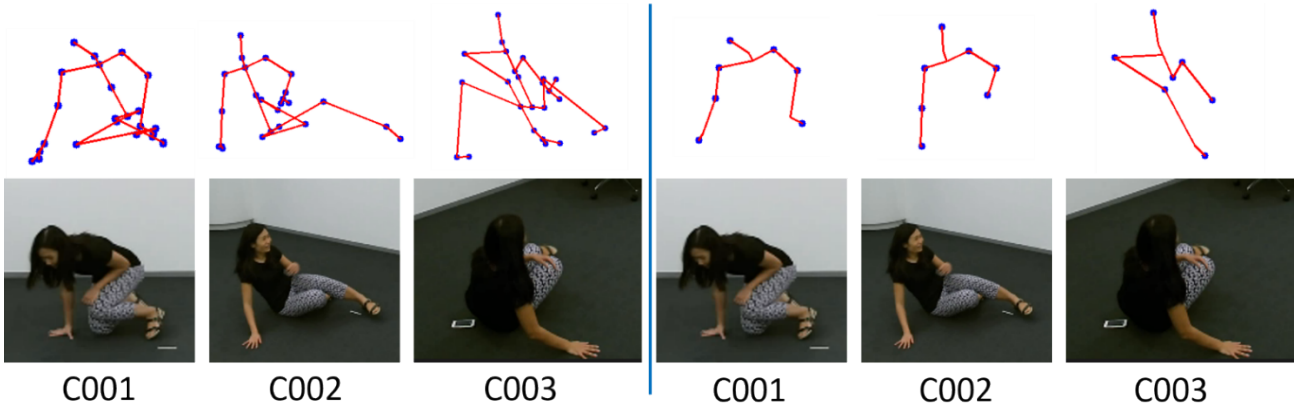

FIGURE 8. The Skeleton and RGB sample from NTU RGB+D dataset. Their sensor uses three angles: −45°, 0°, 45° called C001, C002, C003. The left group shows the original data. The right group is the visualization of our training data.

TABLE 7. Comparison with different fall detection neural network.

	This work	[4]	[5]	[6]	[7]	[8]
Method	Thinning, DNN,	DNN	DNN	DNN	DNN, LSTM	DNN
Dataset	NTU RGB+D	Developed by reference work	Developed by reference work	TST	URFD	KTH
Total Sequence	2722	42	1300	264	70	2391
Network Architecture	MyNet1D-D	MLP	Alex-Net, SSD-Net	3D-CNN	ConvLSTM	Feedback optical flow CNN.
Sensor	Kinect	Kinect	Radar, Optical cameras	Kinect	Kinect	RGB cameras
Input type	Depth image	Depth image	Radar signal, RGB video	Depth image	Depth image	RGB image
Feature	Skeleton	Six important features	TF Feature	Depth image	Depth image	Optical flow
Accuracy	99.2%	98.15%	99.85%	96.9%	98%	98%

We select the current newer vision-based method to compare with our proposed network. These methods are latest research on falling detection, and the accuracy is almost higher than 95% as shown in Table 7. Surely the accuracy is the most important issue, but it is dependent on the method, the number of training data and the model of neural work. Su *et al.* [4] have proposed a neural-network-based fall detection algorithm which uses the feature extraction method to obtain six important features to train the network module. Their proposed network has 98.15% accuracy. However, the dataset is made by themselves. Also the total sequence number is 42 which is not sufficient to prove the generality for their network module.

In [5], a neural network with fusion of multi-sensor information is proposed. It used AlexNet and SSDNet with the radar and optical input data. The accuracy is 99.85% on their developed video sequences. Although the accuracy is higher than ours based on their developed dataset, the network model is larger than our network module as shown in Table 6, meanwhile the computations of AlexNet is 2.5G Ops and the parameter is 60M. As a result, high computations and parameters model is not comfortable on the embedded system. The computations and parameters on our model is smaller than their methods. As a result, it is suitable for the embedded system and can achieve 15 FPS on performance.

In [6], Hwang et al. have presented 3D convolution neural network to analyze the continuous motion data obtained from depth cameras. The accuracy is 96.9% on TST dataset. It can see that our 1D convolution have the similar performance to our work. Additionally, we have considered the pruning technique to reduce excess calculations of neural network.

An end-to-end ConvLSTM is proposed by Abobakr et al. in [7]. The convolution network extracts visual features from input sequence and uses the Long-Shot-Term-Memory (LSTM) to detect fall events. It has 98% accuracy on URFD dataset. A feedback optical flow convolution neural

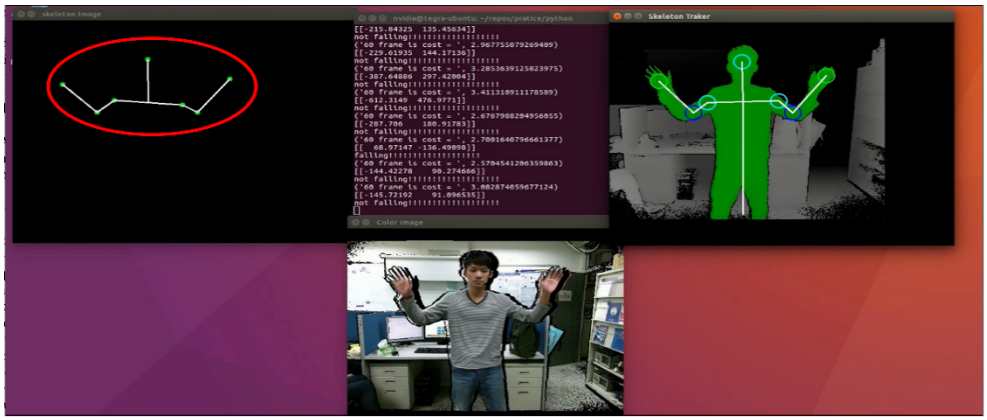

FIGURE 9. Fall detection system on an embedded system.

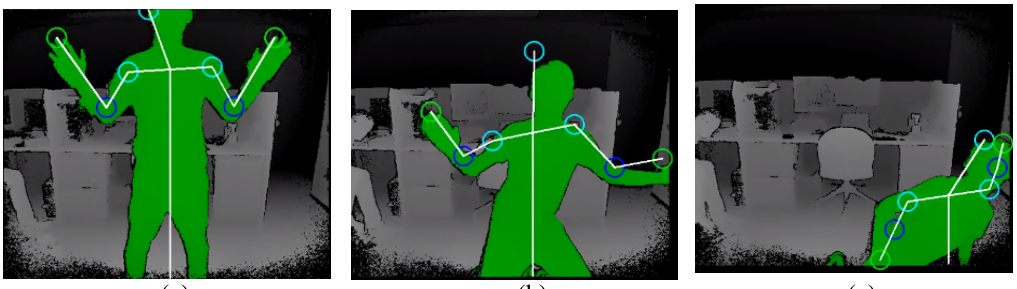

FIGURE 10. Snapshot of fall process (a), (b), (c).

network has the accuracy result of 98%, proposed by Hsieh and Jeng [8]. Our network model is training and testing on the NTU RGB+D witch is the most famous benchmark in the posture recognition technique. It has many challenges on the sequences because the dataset contains 60 actions. It proves that our proposed design can be detecting falling in a wide variety of posture. Also our network model is the smallest one in these designs. Based on this important advantage, we can perform our system on a smaller platform and still have good performance.

B. EMBEDDED SYSTEM DESIGN

For an embedded system design, we have implemented it on NVIDIA Jetson TX2. It contains NVIDIA Pascal GPU which has 256 CUDA core, and the ARM CPU is composed of two ARM v8 64-bit cores. Moreover, this board is able to optimize the multiplication and addition operations of the deep neural network by using CUDA library which results in reducing the calculation time.

The results of the algorithm in the Linux system are shown in Fig. 9, where Linux is running on NVIDIA Jetson TX2. In Fig. 9, top left corner shows the seven nodes which are extracted through the proposed method. These nodes have been highlighted in the red circle. The falling snapshot is shown in Fig. 10. When the network detects the falling case, it will display a red box around the image to indicate that fall event has been detected which is shown in Fig. 10 (c). The system can achieve 15 frames per second for real-time implementation.

VI. CONCLUSION

In this paper, we propose a fall detection system. Our system is based on a vision sensor without using any wearable device to detect fall detection. A depth image has been used to extract the skeleton information and different techniques have been applied to refine the skeleton extraction results. We have modified the skeleton-based method with seven highlight feature points. Moreover, different parameters and the number of calculations have been decreased for the neural network model. Finally, the whole system has been implemented on the NVIDIA Jetson TX2 platform and it has been tested for a real demonstration environment. The system can achieve 15 frames per second for real-time implementation. Based on this non-wearable falling detection system, it can be efficiently realized in some health-care applications. In future work, we will reduce the hardware cost, such as change the Kinect to a normal camera as input. In addition, we will increase the versatility of system on posture recognition. Then our system can not only detect the falling but also understood other postures. Finally, it can achieve a low cost and high accuracy posture recognition system.

ACKNOWLEDGMENTS

The research in this paper used the NTU RGB+D Action Recognition Dataset made available by the ROSE Lab at the Nanyang Technological University, Singapore.

REFERENCES

[1] J. Murphy and B. Isaacs, "The post-fall syndrome: A study of 36 elderly patients," *Gerontology*, vol. 28, no. 4, pp. 265–270, 1982.

[2] N. Noury, P. Rumeau, A. Bourke, G. ÓLaighin, and J. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, no. 6, pp. 340–349, Dec. 2008.

[3] M. Vallejo, C. V. Isaza, and J. D. López, "Artificial neural networks as an alternative to traditional fall detection methods," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 1648–1651.

[4] M.-C. Su, J.-W. Liao, P.-C. Wang, and C.-H. Wang, "A smart ward with a fall detection system," in *Proc. IEEE Int. Conf. Environ. Elect. Eng. IEEE Ind. Commercial Power Syst. Eur. (EEEIC / I&CPS Europe)*, Jun. 2017, pp. 1–4.

[5] X. Zhou, L.-C. Qian, P.-J. You, Z.-G. Ding, and Y.-Q. Han, "Fall detection using convolutional neural network with multi-sensor fusion," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2018, pp. 1–5.

[6] S. Hwang, D. Ahn, H. Park, and T. Park, "Poster abstract: Maximizing accuracy of fall detection and alert systems based on 3D convolutional neural network," in *Proc. IEEE/ACM 2nd Int. Conf. Internet-Things Design Implement. (IoTDI)*, Apr. 2017, pp. 343–344.

[7] A. Abobakr, M. Hossny, H. Abdelkader, and S. Nahavandi, "RGB-D fall detection via deep residual convolutional LSTM networks," in *Proc. DICTA*, Dec. 2018, pp. 1–7.

[8] Y.-Z. Hsieh and Y.-L. Jeng, "Development of home intelligent fall detection IoT system based on feedback optical flow convolutional neural network," *IEEE Access*, vol. 6, pp. 6048–6057, 2017.

[9] T.-T. Nguyen, M.-C. Cho, and T.-S. Lee, "Automatic fall detection using wearable biomedical signal measurement terminal," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 5203–5206.

[10] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Med. Eng. Phys.*, vol. 30, no. 1, pp. 84–90, 2008.

[11] A. Jefiza, E. Pramunanto, H. Boedinoegroho, and M. H. Purnomo, "Fall detection based on accelerometer and gyroscope using back propagation," in *Proc. 4th Int. Conf. Elect. Eng., Comput. Sci. Inform. (EECSI)*, Sep. 2017, pp. 1–6.

[12] X. Yuan, S. Yu, Q. Dan, G. Wang, and S. Liu, "Fall detection analysis with wearable MEMS-based sensors," in *Proc. 16th Int. Conf. Electron. Packag. Technol. (ICEPT)*, Changsha, China, Aug. 2015, pp. 1184–1187.

[13] G. M. Basavaraj and A. Kusagur, "Vision based surveillance system for detection of human fall," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 1516–1520.

[14] W.-Y. Shieh and J.-C. Huang, "Speedup the multi-camera video-surveillance system for elder falling detection," in *Proc. Int. Conf. Embedded Softw. Syst.*, May 2009, pp. 350–355.

[15] Y.-S. Lee and H. Lee, "Multiple object tracking for fall detection in real-time surveillance system," in *Proc. 11th Int. Conf. Adv. Commun. Technol.*, Feb. 2009, pp. 2308–2312.

[16] A. Manzi, L. Fiorini, R. Limosani, P. Dario, and F. Cavallo, "Two-person activity recognition using skeleton data," *IET Comput. Vis.*, vol. 12, no. 1, pp. 27–35, Feb. 2018.

[17] T. Jia, Y. Zhang, R. Liu, and H. Wang, "Face authentication method based on Kinect device," in *Proc. IEEE Int. Conf. Cyber Technol. Automat., Control, Intell. Syst. (CYBER)*, Jun. 2016, pp. 179–183.

[18] T. Liu, Y. Song, Y. Gu, and A. Li, "Human action recognition based on depth images from microsoft kinect," in *Proc. 4th Global Congr. Intell. Syst. (GCIS)*, Dec. 2013, pp. 200–204.

[19] T. Q. Vinh and N. T. Tri, "Hand gesture recognition based on depth image using Kinect sensor," in *Proc. 2nd Nat. Foundation Sci. Technol. Develop. Conf. Inf. Comput. Sci. (NICS)*, Sep. 2015, pp. 34–39.

[20] Y. Hbali, S. Hbali, L. Ballihi, and M. Sadgal, "Skeleton-based human activity recognition for elderly monitoring systems," *IET Comput. Vis.*, vol. 12, no. 1, pp. 16–26, Feb. 2018.

[21] S. C. Agrawal, R. K. Tripathi, and A. S. Jalal, "Human-fall detection from an indoor video surveillance," in *Proc. Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2017, pp. 1–5.

[22] P. S. Sase and S. H. Bhandari, "Human fall detection using depth videos," in *Proc. Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2018, pp. 546–549.

[23] A. S. Rao, J. Gubbi, S. Marusic, and M. Palaniswami, "Crowd event detection on optical flow manifolds," *IEEE Trans. Cybern.*, vol. 46, pp. 1524–1537, Jul. 2016.

[24] T. Wang and H. Snoussi, "Detection of abnormal visual events via global optical flow orientation histogram," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 988–998, Jun. 2014.

[25] S. S. Sengar and S. Mukhopadhyay, "A novel method for moving object detection based on block based frame differencing," in *Proc. Int. Conf. Recent Adv. Inf. Technol. (RAIT)*, Mar. 2016, pp. 467–472.

[26] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.

[27] Y. Tian, A. Senior, and M. Lu, "Robust and efficient foreground analysis in complex surveillance videos," *Mach. Vis. Appl.*, vol. 23, no. 5, pp. 967–983, 2012.

[28] M. Ullah, H. Ullah, N. Conci, and F. G. B. De Natale, "Crowd behavior identification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 1195–1199.

[29] M. Ullah and F. A. Cheikh, "Deep feature based end-to-end transportation network for multi-target tracking," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 3738–3742.

[30] K. Jaganathan, S. K. Panagiotopoulou, J. F. McRae, S. F. Darbandi, D. Knowles, Y. I. Li, J. A. Kosmicki, J. Arbelaez, W. Cui, G. B. Schwartz, E. D. Chow, E. Kanterakis, H. Gao, A. Kia, S. Batzoglou, S. J. Sanders, and K.-K.-H. Farh, "Predicting splicing from primary sequence with deep learning," *Cell*, vol. 176, pp. 535–548, Jan. 2019.

[31] L. Alhimale, H. Zedan, and A. Al-Bayatti, "The implementation of an intelligent and video-based fall detection system using a neural network," *Appl. Soft Comput.*, vol. 18, pp. 59–69, May 2014.

[32] Z. Huang, Y. Liu, Y. Fang, and B. K. P. Horn, "Video-based fall detection for seniors with human pose estimation," in *Proc. Int. Conf. Univ. Village (UV)*, Oct. 2018, pp. 1–4.

[33] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. CVPR*, Jul. 2017, pp. 7291–7299.

[34] M. S. Allili, N. Bouguila, and D. Ziou, "A robust video foreground segmentation by using generalized Gaussian mixture modeling," in *Proc. 4th Can. Conf. Comput. Robot Vis.*, May 2007, pp. 503–509.

[35] T. Kryjak and M. Gorgon, "Real-time implementation of the ViBe foreground object segmentation algorithm," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2013, pp. 591–596.

[36] T.-H. Tsai and C.-H. Chang, "Hardware/software co-design and VLSI implementation for the intelligent surveillance system," *IEEE Sensors J.*, vol. 17, pp. 6077–6089, 2017.

[37] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984.

[38] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Commun. ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[39] M. Kumar and S. Singh, "Edge detection and denoising medical image using morphology," *Int. J. Eng. Sci. Emerg. Technol.*, vol. 2, no. 2, pp. 66–72, 2012.

[40] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1010–1019.

[41] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. CVPR*, Jun. 2015, pp. 1110–1118.

[42] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI*, 2016, pp. 1–7.

[43] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.

TSUNG-HAN TSAI received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1990, 1994, and 1998 respectively. From 1999 to 2000, he was an Associate Professor of electronic engineering with Fu Jen University. In 2000, he joined National Central University, where he has been a Full Professor with the Department of Electrical Engineering, since 2008. He is currently the Director of Intelligent Chip and System Center, National Central University, and also serves as the Principal Investigator of the National Program for Intelligent Electronics. He has been awarded more than 40 patents and 230 refereed articles published in international journals and conferences. His research interests include VLSI signal processing, video/audio coding algorithms, DSP architecture design, wireless communication, and system-on-chip design. He has been an IEEE member for more than 15 years. He serves as a Technical Program Committee Member or the Session Chair of several international conferences. He was a recipient of the Industrial Cooperation Award, in 2003, from the Ministry of Education, Taiwan. He was also a recipient of the Best Article Award from the IEEE International Conference on Innovations in Bio-inspired Computing and Applications (IBICA), in 2011, and the IEEE International Conference on Innovation, Communication and Engineering (ICICE), in 2015. His research team has won many international IC-related student design contest awards, including ISSCC, in 2011, ISOCC, in 2015, and TI DSP Asia Design Contest, in 2008. He was the General Co-Chair of the IEEE International Conference on the Internet of Things, in 2014. He has served as the Guest Editor of special issues for the *Journal of VLSI Signal Processing Systems*.

CHIN-WEI HSU is currently pursuing the M.S. degree in electrical engineering from National Central University, Taiwan. His research interests include image processing, deep learning, and embedded system design.