JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 34, 577-591 (2018)
DOI: 10.6688/JISE.201805_34(3).0001

Convolutional Recurrent Neural Networks for Posture Analysis in Fall Detection*

HSIU-YU LIN[1], YU-LING HSUEH[1,3] AND WEN-NUNG LIE[2,3]
[1]*Department of Computer Science and Information Engineering*
[2]*Department of Electrical Engineering*
[3]*Center for Innovative Research on Aging Society (CIRAS)*
National Chung Cheng University
Chiayi, 621 Taiwan
E-mail: {lhy104m; hsueh}@cs.ccu.edu.tw; ieewnl@ccu.edu.tw[2]

Existing methods have extensively addressed the issue of detecting abnormal events in a smart home environment through wearable sensors in the past years. However, the limitations of wearable sensors include the limited battery power as well as the use and adoption challenges of wearable activities on a daily basis. The use of non-wearable and non-intrusive sensors is necessary for providing better user experiences and achieving a sustainable and reliable detection model. However, it is still very challenging to analyze such non-wearable sensor data with a high level of accuracy. In this paper, we present a continuous deep learning model which receives a set of consecutive images for classifying posture types using a Microsoft Kinect as our non-wearable sensor. We adopt a deep learning technique called the recurrent neural network (RNN) using the long short-term memory (LSTM) architecture to construct our detection model by identifying human postures in fall detection. Furthermore, we investigate the inputs for our model by extracting the features from the pre-processed high-resolution RGB images, including body shape, depth and optical flow. As a result, the body shape with genuine motion and depth information are considered. Finally, we present the experimental results to demonstrate the performance and novelty of our approach.

Keywords: posture analysis, convolutional neural network, long short-term memory network, deep learning, fall detection

1. INTRODUCTION

In recent years, the population aged 60 and above has increased significantly, with the number now exceeding 600 million worldwide [1]. Because of the explosive growth in the elderly population, the financial burden and pressure on the younger and middle-aged population (from age 15 to 64) has become extremely heavy. Due to the deteriorating physical health of the elderly, one of the critical issues is to monitor their daily activities to ensure their safety. In the event of accidents, they must be assisted; otherwise, it may lead to serious injury or even death. One feasible way to ensure the safety of the elderly is to accommodate them in a nursing home. However, the cost is high and relocating to such an unfamiliar place might be traumatic for many. On the other hand, if a senior adult lives at home alone, it is difficult to receive immediate assistance in the event of an emergency. Therefore, a low-cost, light-weight system to address this issue is presented in this paper. To reduce the time in which senior adults can receive help when accidents happen in the home environment, we have implemented a real-time abnormal

Received February 7, 2017; revised March 22, 2017; accepted April 22, 2017.
Communicated by Ren-Hung Hwang.
* The preliminary version was appeared in International Computer Symposium (ICS) 2016, Taiwan.

event detection system based on the posture analysis to precisely detect such events.

Several studies have worked on fall detection in the home environment. Some of these approaches have adopted a wearable sensor such as a smart watch, which has become popular in recent years. Although a smart watch is now low-cost and easy to obtain, it still presents numerous problems. For example, the battery life is limited, and seniors often resist using new high-tech products. In this paper, we propose to automatically detect abnormal events at home without wearable sensors. For those who need to work in the daytime, one of their most significant concerns is what will happen to their senior relatives at home. The feature of instant notification is essential when a senior is in sudden danger so that they can receive the necessary assistance as soon as possible. Based on the above-mentioned issues, we have created a method based on deep learning to construct an abnormal event detection model. We mainly focus on the posture analysis in fall detection to automatically recognize abnormal events in the home. If any abnormal event happens, one can be informed immediately by the system.

Deep Learning has become a popular method in machine learning in recent years and has been extensively used in many studies [6-24]. It is an architecture incorporating a multi-layer neuron network. In 1989, LeCun *et al.* used a backpropagation algorithm in a deep neural network [10] which made a huge step forward in machine learning. Although the algorithm had achieved good performance, the training processes took a considerable amount of time. Fortunately, due to the advanced hardware and algorithms, the performance has been greatly improved during the past two decades. Furthermore, the restricted Boltzmann machine (RBM) and deep learning networks have been combined to achieve unsupervised learning. Deep learning with high classification accuracy and efficiency is now widely used, for example, by YouTube and Facebook. In this study, we used a Kinect as the sensor to collect the data in the environment, because it can capture three major data streams. First, the skeletal joint stream is used for collecting data on 26 joints in the body. Second, the depth stream is used for measuring the depth between an object and a sensor. Third, the RGB image stream is used for recording the activities of the seniors.

In this paper, we use both RGB images and depth images collected by a Kinect. We preprocess the RGB images which are subsequently input to the deep learning model so that the depth images are used directly. We apply a convolutional neural network and long short-term memory to labeled images, and evaluate the trained model for posture classification. For the experiments described in Section 5, we first compare the detection results using the inputs of fusion images with various features including body shape, depth, and optical flow for our model with the same layers and neurons. Second, we evaluate the results by varying the layers and neurons for the same inputs. Finally, we analyze the results using the inputs with various features extracted from the RGB images for our model to show that our proposed method outperforms existing methods. The experiment results show that our proposed method has achieved better accuracy.

2. RELATED WORK

A number of researchers have focused on fall detection by utilizing environmentally mounted sensors, such as infrared sensors, conventional cameras, and depth imaging

sensors. A Microsoft Kinect has an infrared sensor, a depth image sensor, and a camera all in one. The detection range of a Kinect is from 0.5 to almost 7 meters, which is an ideal range for home detection. A number of technologies [2-5] exist for the purpose of event detection using a Kinect. In [2], the researchers used a Kinect for monitoring intake gestures on a dining table. Parra-Dominguez *et al.* [2] proposed a fall detection model that monitors the indoor stairs to capture the skeletal joint data for measurement.

There have been intensive studies on developing the architectures in deep learning. The four most popular methods are deep neuron networks (DNNs), convolutional neural networks (CNNs), deep belief networks (DBNs), and convolutional deep belief networks (CDBNs). DNN is a multi-layer neuron network, and is the most original method. However, it incurs significant computational time [6]. CNN [8-13, 20] is a multi-layer NN, in which every layer consists of multiple two-dimensional figures. There are two types of layers in CNN, the convolutional and the subsampling layers. Every subsampling layer is implemented after a convolutional layer. Images are used to strengthen the features and reduce the noises after every convolutional layer. On every subsampling layer, the images can be narrowed, but they still retain the useful features. Deep belief networks (DBNs) are deep neuron networks with a restricted Boltzmann machine (RBM) in the first layer, while the other layers are Bayesian networks. Hayat *et al.* presented a DBN structure on image reconstruction [6] for the incomplete picture of a person's face, which further identifies the face image. Lee, Honglak, *et al.* presented CDBN, a deep learning architecture for unsupervised two-dimensional image classification [7].

Since 2012, deep learning has drawn extensive attention after Krizhevsky *et al.* [8] presented a deep convolutional neural network with a new activation function and a parallel implementation training process on multiple GPUs to enhance its efficiency. The proposed approach outperformed most of the existing machine learning techniques and showed the power of deep learning in object recognition. The approaches in [10-12] have modified the convolutional neural network and obtained higher accuracy. A famous event in 2016 was that the artificial intelligence model, AlphaGo, achieved great success in beating a human professional Go player. Tian *et al.* [13] applied a combination model of CNN and the Monte Carlo search tree to improve the accuracy of determining the best prediction of all the choices. Deep learning has also accomplished great success in audio inputs. Hinton *et al.* [15] implemented a deep recurrent neural network for speech recognition. In [16, 17], a CNN was used to estimate various types of postures. Donahue, Jeffrey, *et al.* [20] demonstrated a combined deep learning model based on a CNN for visual feature extraction and long short-term memory for continuous classification. Unlike a traditional CNN which can only detect a single object in an image, image segmentation [23, 24] is an extension of CNN and is also known as R-CNN which is used to detect multiple objects in the images and is more powerful than the traditional CNN.

3. PRELIMINARY

3.1 Deep Neural Networks (DNNs)

Deep learning has drawn extensive attention in recent years and has been widely used in many fields. The deep learning method is also referred to as deep neural networks (DNNs), as it consists of multiple layers, each of which consists of multiple neu-

rons that compute a specified activation function. Deep learning achieves high accuracy because of the modularization of the model. When a DNN has only one layer and is given insufficient training data, the model might not achieve good results. In this paper, we therefore first train the model by using the inputs of fusion images with features retrieved from a large training data set, and similarly, use the fusion images with the features retrieved from the small training data set for the next layer so that the model can achieve better results.

Deep neural networks were the earliest method presented in deep learning. In recent years, deep learning has become a huge trend in artificial intelligence. Much research has implemented different types of structures, all of which are based on deep neural networks.

3.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are one of the most well-known types of architecture in DNNs. CNNs mostly process two-dimensional data such as images, and use these data to train a model. CNNs include two components: feature maps and multiple layers in the neural network. Feature maps are divided based on the input data. These feature maps are used in the convolutional layers to output more important features, and in the subsampling layers to enlarge the features after the processes in every convolutional layer are completed. The second part of CNNs is the multi-layer of a neural network which is used to predict the final results. Convolutional neural networks are known for their powerful visual feature extraction. Unlike deep neural networks, CNNs use fewer weighting factors and increase the pixel correlation of the images in the network, thus obtaining good performance. In our method, we use a CNN for video feature extraction, where the inputs are fusion images, and the outputs are multiple feature maps.

3.3 Recurrent Neural Networks (RNNs)

Recurrent neural networks utilize the internal memory to "remember" the previous results and compare them with the inputs. The final neurons maintain the information of the entire network. Although RNNs have been proven to be effective in many studies, it is difficult to train a model to learn long-term dynamics, as they may suffer from the vanishing and exploding gradients problem. Long short-term memory networks (LSTMs) [15] are an extension method for the vanishing gradient descent problem in recurrent neural networks. A solution was provided by incorporating memory units which can learn when to ignore a previous state or when to update the state. Each LSTM neuron contains three gates: an input gate, a forget gate and an output gate. Each gate learns when to open or close by using the sigmoid function during the training process. The memory cell unit focuses on two things: the previous memory unit and the current input as well as the previous hidden state. The input gate and the forget gate are sigmoidal, which means their values fall within the range [0,1]. Therefore, the RNNs can selectively forget the previous memory or consider the current input. The output gate learns the number of the memory cells to transfer to the hidden state. With these gates, LSTMs are able to learn long-term and extremely complicated information. The only shortcoming is that LSTMs incur significant training time because one unit requires intensive calcula-

tions. Because of the huge improvement in the recurrent neural network by LSTM, RNN has become a powerful method to build a model with sequential inputs. Therefore, in our model, we implement RNNs that receive the flattened CNN outputs of video inputs.

4. DEEP ABNORMAL POSTURE DETECTION MODEL

We used a Microsoft Kinect version 2 as our sensor, because it has a high resolution RGB camera and better detection range from 0.5 to 7 meters than version 1. A Kinect can take three types of images: RGB images, depth images, and infrared images. A Kinect can also record audio data and show up to 6 people with 25 joint points.

Our system structure of this research is shown in Fig. 1. The first part is the image processing layer where the inputs are videos and the outputs are black and white body shapes. We collect RGB images and transform them into body-shaped and optical flow images. The depth images are received simultaneously along with the RGB images. The second part is a deep learning structure, which consists of a convolutional neural network (CNN) for visual feature extraction and a recurrent neural network (RNN) for keeping track of the information of the previous outputs. Finally, the deep learning model is built to classify the postures in fall detection.

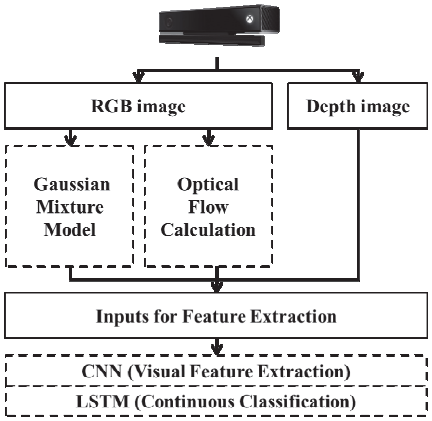


Fig. 1. System architecture.

4.1 Training Images

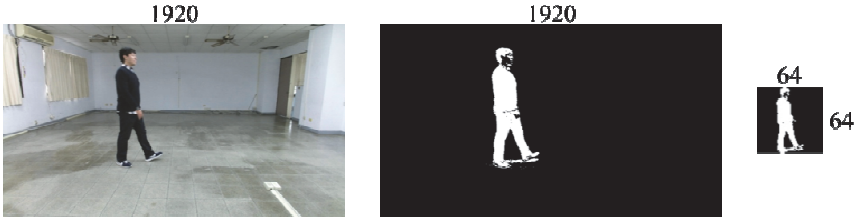
The Gaussian mixture model (GMM) RGB images are processed by our image processing algorithm after we obtain the image streams from a Kinect. GMM is a probabilistic model for presenting the presence of subpopulations. In our work, several background images are used to build a GMM model. A Gaussian mixture model has three parameters, namely mixture weights (w_i), mean vector (μ_i) and covariance matrix ($\sum_i$). M means the amount of Gaussian distribution and λ is the GMM for each pixel. If our data $X_N = \{X_1, X_2, \ldots, X_n\}$ is distributed in D dimensions, the probability of GMM is given by:

$$\lambda = \{w_i, \mu_i, \sum_i\}, i = 1, 2, \ldots, M,$$

$$p(X_N \mid \lambda) = \sum_{i=1}^{M} w_i g_i(X_N),$$

$$g_i(x) = \frac{1}{(2\pi)^{D/2} \mid \sum_i \mid^{1/2}} e^{D} \qquad D = -\frac{1}{2}(x-\mu_i)^T \sum_i^{-1} (x-\mu_i).$$

As a result, we can subtract the foreground from the background images. Subsequently, we further implement a sequence of image processing procedures to acquire a fixed-resolution body shape for each image. During the image processing procedures, we apply the erosion and dilation filters to fill the empty holes in the foreground first. Next, shadow elimination is implemented to conserve the body shape. In order to keep the features in the image, we apply the body shape detection. Finally, we shrink each output of the detected body shape into a fixed resolution and shift it to the center of the image. Fig. 2 shows the results of the GMM model and the image shifting and shrinking. We have tested our GMM given both a simple and a complex background. For both situations, the GMM model results in high-quality extraction of the human body shapes.



(a) (b) (c)

Fig. 2. (a) Original image [1920×1080]; (b) The result of background subtraction using the GMM model [1920×1080]; (c) The output of image shrinking and shifting [64×64].

Optical flow In order to detect human movement, we obtain optical flow images in our research. We compare the difference between two continuous frames, and calculate the optical flow images based on the method in [14]. Lucas *et al.* [14] presented a method which can calculate optical flow using multiple pyramids. Two different images are processed as outputs with grey scale: one for the motion on the x axis, the other for the motion on the y axis. Each pixel presents the motion between the previous frame and the current frame. We transform the values in the matrix and store them in a grayscale image. An example output of the optical flow calculation is shown in Fig. 3. A darker pixel represents a negative value while the lighter one represents a positive value. Finally, we implement body shape detection to extract the body shape and shrink the optical flow images into a fixed size.

Depth image We use a Microsoft Kinect version 2 as our sensor, and Kinect APIs to capture both RGB images and depth images at the same time. The resolution of the depth images is 512×424. Depth images are shown in gray scale, where the closer to the sensor, the darker the pixel. We can therefore know the distance between the sensor and the objects. Similar to the body shape images and optical flow images, we also apply body detection to capture the depth images with only the body shape. Fig. 4 shows an example of a depth image.

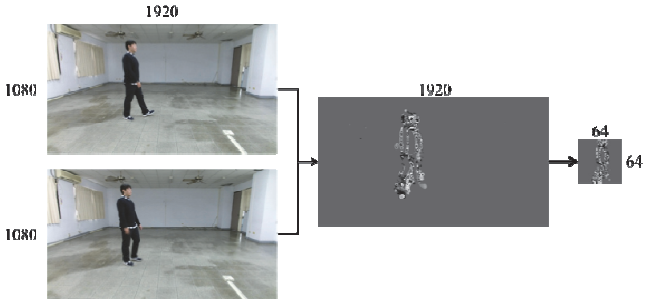


Fig. 3. (a) Original image [1920×1080]; (b) The output of the optical flow calculation [1920×1080]; (c) The output of image shifting and shrinking [64×64].

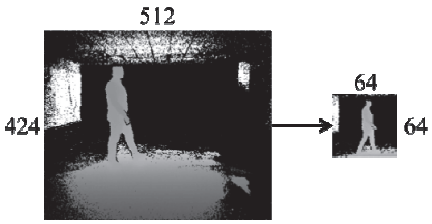


Fig. 4. (a) Original depth image [512×424]; (b) The result of image shifting and shrinking [64×64].

Finally, based on the above-mentioned procedures, we obtain the inputs of fusion images with three features: the body shape, optical flow and depth. All these images are shrunk into 64×64 pixels. We reposition each of these images into a 64×64 fusion image. Fig. 5 shows the output image after merging these three types of features.

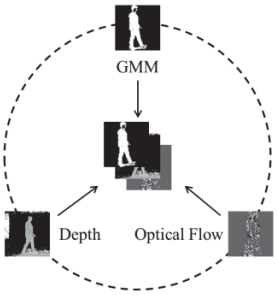


Fig. 5. An example of a fusion image.

4.2 Model Architecture and Configurations

Fig. 6 shows the structure of our model. We combine the convolutional neural networks (CNNs) and the recurrent neural networks (RNNs). As the image processing procedures are described in Section 4.1, we transform the RGB images into body shape and optical flow images. Given these images as inputs, the CNN extracts the visual features

from these images after multiple layers of convolutional layers and subsample layers are performed. In our model, we use multiple RNN layers that receive the flattened CNN outputs of the feature maps. The RNN layers consist of multiple LSTMs, which remember the previous results to compute the current results. All the weights and bias in the model are shared to avoid increasing weights. We assume that this model is spatially and temporally deep. Because the system sequential receives the input images, we can only build one model. Every input leads to updating the variables while training a model. When testing our proposed model, we use the inputs of the RGB and depth images and perform the processes described in Section 4.1 to obtain fusion images. The goal of the model is to identify a type of posture in a given fusion image.

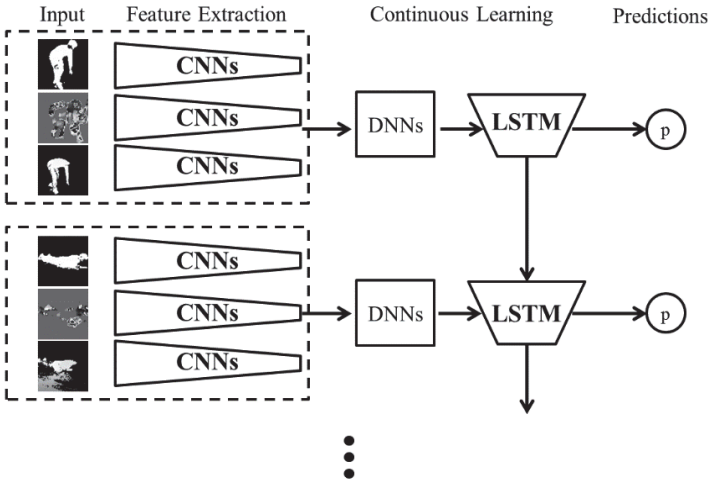


Fig. 6. Our proposed deep learning structure.

We setup a room to simulate a smart home environment for the experiments. A Kinect was installed on a table at a height that can capture the user's head and feet clearly. We captured 7,500 RGB and depth images in total for training our model. Because both the RGB and depth images were too large to train, we resized the RGB images from 1920*1080 to 64*64 and the depth images from 512*424 to 64*64 as described in Section 4.1. We defined an eight-layer deep learning model to detect the postures in fall detection, as shown in Fig. 7. The first convolutional layer contains 32 filters with 3×3 filter size and sets the stride as 1. The 32 feature maps are the next inputs for maxpooling to reduce dimensions with the same stride. For the second convolutional layer and the maxpooling layer, we implemented 64 3×3 filters and set the stride as 1. We used 64 filters to obtain 64 feature maps and also used a maxpooling layer to reduce the number of dimensions. The third layer is also the convolution layer and maxpooling layer with 128 filters. The flattened layer was implemented after the three layers of convolution and maxpooling. In the last five layers, we implemented one layer of LSTMs in the seventh layer. Finally, we classify the inputs into four categories, namely background (*i.e.*, nobody in the image), walking, lying and falling. The details of the experimental results are described as follows.

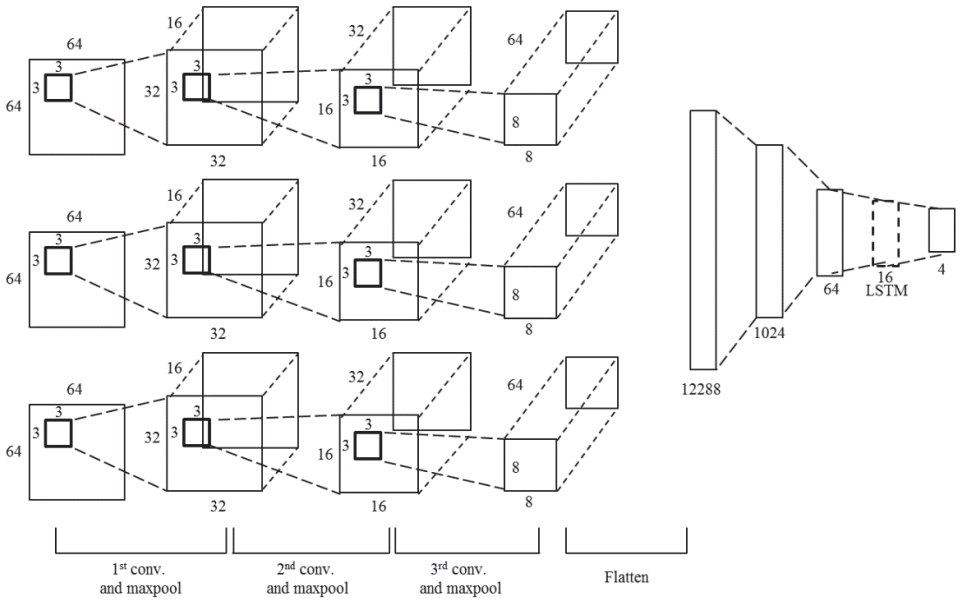


Fig. 7. Deep learning model.

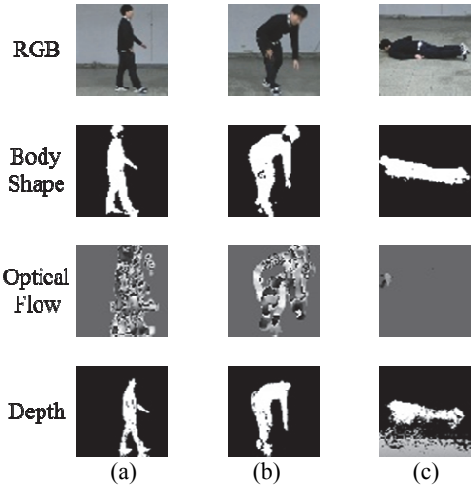
5. EXPERIMENTAL EVALUATION

In this research, we used an Intel Core i7-2600 CPU and NVIDIA GeForce GTX 970 with CUDA version 7.5 to speed up the training of the convolutional neural network and the long short-term memory network. For the image processing procedures, we used openCV on C++. Theano and Keras were used as the toolkit for training and testing all of the data. The training time was one minute per epoch and we trained 100 epochs.

In this section, we present the experimental results of our deep learning model. In this experiment, four types of images are used as our inputs. That is, body shape, depth, optical flow, and fusion images with all of these three features. Specifically, the body shape, depth and optical flow images are 64×64 images with only one channel; the fusion images are also 64×64 images with three channels. We used an eight-layer deep model with one dropout layer after every maxpooling layer. The parameters used throughout the experiments are shown in Table 1. The dropout rate is the probability of the neurons used in the model dropping out while training. For the activation function, we chose ReLU instead of sigmoid. For loss function and optimizer we adopted cross entropy and RMSprop, respectively, in this model. The dataset contains 7,500 RGB and depth images, respectively, and among them, 500 of the images are background images where no one is present. We used these images to build the GMM model. We used 4,900 images (70% of the data set) to train our model, and 2,100 images (30% of the data set) to test the model. The last 7,000 images contain four categories, namely walking, falling, lying and background as shown in Fig. 8. Optical flow image shows the motion between two frames. If the user does not move, the optical flow value us zero. The left side of the figure represents the walking category, the middle represents the falling category, and the right side represents the lying category.

Table 1. Experimental settings.

Dropout	Activation Function	Loss function	Optimizer	Batch size	Epoch
0.4	ReLU	cross entropy	RMSprop	128	100



(a) (b) (c)

Fig. 8. (a) Walking; (b) Falling; (c) Lying.

5.1 Input Comparison

In this section, we compare the inputs of fusion images with various features. As we describe in Section 4, we design an algorithm to receive a 64×64, three-channel input of a fusion image with three features including body shape, optical flow and depth. Optical flow is the pattern of motion of an object in two vector matrixes. These vector matrixes describe the moving vector on the x axis and the y axis. We compare these two features of the images, and the results are shown in Fig. 9. As we can see from the figure, the optical flow on the y axis outperforms the optical flow on the x axis. This consequence is expected as the moving on the y axis is likely to represent the falling event better than the x axis.

Next, we examine our inputs of fusion images with various features. We compare inputs including body shape, the optical flow of the y axis and depth images with the inputs of RGB images. The results are shown in Fig. 10. As we can see from the figure, the fusion images used as inputs achieve the best accuracy. The result indicates that fusion images containing the features extracted from the RGB images facilitate the classification processes for our model so as to achieve better accuracy than the original RGB images.

5.2 Model Comparison

In this section, we compare our deep learning model with LSTM in the seventh layer with the pure convolutional neural network (CNN) of an existing model called *Image-*

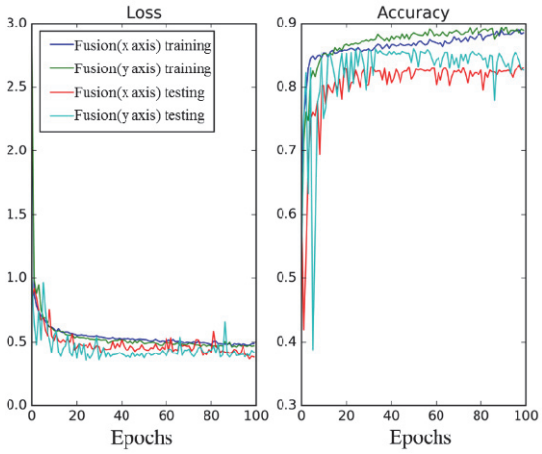


Fig. 9. The effect of optical flow on the x axis and the y axis.

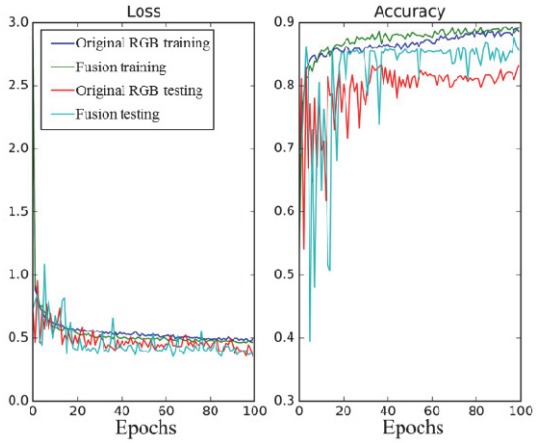


Fig. 10. Effect of inputs of fusion images vs. original RGB images.

net in [8]. The difference between our model and the CNN model of *imagenet* is that in the seventh layer, we use LSTM instead of a neural network to capture the correlation from the consecutive image inputs. Because the falling action is continuous, we consider the value from the previous iteration to classify the current state. Fig. 11 shows the results of *Imagenet* and our proposed model. As we can see from the figure, our deep learning model with LSTM outperforms *Imagenet*. The result shows that the LSTM model is able to capture the motion pattern from the consecutive images so as to achieve better classification results.

In terms of the CPU performance, the training time for *Imagenet* is 25 seconds per epoch and our proposed model is 60 seconds per epoch. The average testing time per input is 1.42 milliseconds for *Imagenet* and 2.38 milliseconds for our proposed model. Our performance is slightly slower than that of *Imagenet*; however, the running time is still efficient enough to support real-time detection.

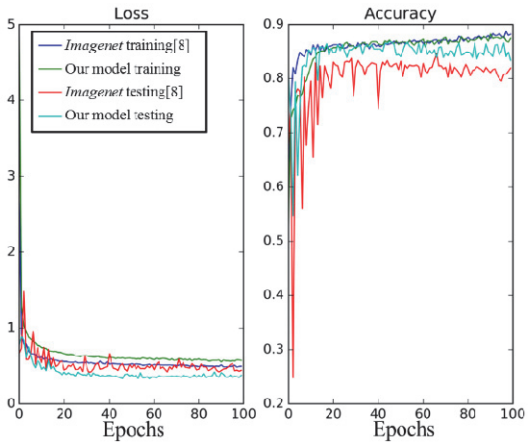


Fig. 11. Effect of various models on loss and accuracy.

6. CONCLUSION

In this paper, we present a deep learning classification model for postures in fall detection by using a Kinect as the sensor. Our deep learning model utilizes the convolutional neural network for visual feature extraction and the long short-term memory network for classification.

We investigate the inputs of fusion images for our model by extracting the features from the pre-processed high-resolution RGB images including the body shape, depth, and optical flow. The experimental results show that our proposed model is efficient and achieves better classification results than the existing model. Therefore, our model can support real-time posture recognition in fall detection.

REFERENCES

1. H. M. Hondori, M. Khademi, and C. V. Lopes, "Monitoring intake gestures using sensor fusion (microsoft kinect and inertial sensors) for smart home tele-rehab setting," in *Proceedings of the 1st Annual IEEE Healthcare Innovation Conference*, 2012, pp. 36-39.
2. G. S. Parra-Dominguez, B. Taati, and A. Mihailidis, "3D human motion analysis to detect abnormal events on stairs," in *Proceedings of the 2nd IEEE International Conference on 3D Imaging*, *Modeling*, *Processing*, *Visualization and Transmission*, 2012, pp. 97-103.
3. E. E. Stone and M. Skubic, "Unobtrusive, continuous, in-home gait measurement using the Microsoft Kinect," in *Proceedings of IEEE Transactions on Biomedical Engineering*, Vol. 60, 2013, pp. 2925-2932.
4. E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," *IEEE Journal of Biomedical and Health Informatics*, Vol. 19, 2015, pp. 290-301.
5. S. Jankowski, *et al.*, "Deep learning classifier for fall detection based on IR distance

sensor data," in *Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Vol. 2, 2015, pp. 723-727.
6. M. Hayat, M. Bennamoun, and S. An, "Deep reconstruction models for image set classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, 2015, pp. 713-727.
7. H. Lee, *et al.*, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th ACM Annual International Conference on Machine Learning*, 2009, pp. 609-616.
8. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105.
9. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, Vol. 1, 1989, pp. 541-551.
10. A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," Unpublished manuscript 40, 2010.
11. H. Lee, *et al.* "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in Neural Information Processing Systems*, Vol. 22, 2009, pp. 1096-1104.
12. A. Jain, *et al.*, "Modeep: A deep learning framework using motion features for human pose estimation," in *Proceedings of the 12th Asian Conference on Computer Vision*, 2014, pp. 302-315.
13. Y. Tian and Y. Zhu, "Better computer go player with neural network and long-term prediction," in *Proceedings of International Conference on Learning Representations*, arXiv preprint arXiv:1511.06410, 2015,
14. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vol. 2, 1981, pp. 674-679.
15. A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645-6649.
16. J. Tompson, *et al.*, "Efficient object localization using convolutional networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648-656.
17. J. Tompson, *et al.*, "Joint training of a convolutional network and a graphical model for human pose estimation," *Advances in Neural Information Processing Systems*, 2014, pp. 1799-1807.
18. M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," arXiv preprint arXiv:1511.05440, 2015.
19. S. Kolouri and G. K. Rohde, "Transport-based single frame super resolution of very low resolution face images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4876-4884.
20. J. Donahue, *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, arXiv preprint arXiv:1411.4389, 2014.

21. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
22. Y. Wu, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," arXiv preprint arXiv:1609.08144, 2016.
23. R. Girshick, *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580-587.
24. S. Ren, *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, 2015, pp. 91-99.

Hsiu-Yu Lin (林修宇) received his BS and MS degrees in Computer Science and Information Engineering from the Kaoshiung Medical University in 2012, and from the National Chung Cheng University in 2016, respectively. His research interests include machine learning, neural network and deep learning.

Yu-Ling Hsueh (薛幼苓) is an Associate Professor with the Department of Computer Science and Information Engineering at the National Chung Cheng University, Taiwan since 2011. Hsueh received her M.S. and Ph.D. degrees in Computer Science from the University of Southern California in 2003 and 2009, respectively. Her research interests are spatio-temporal databases, mobile data management, scalable continuous query processing, and spatial data indexing.

Wen-Nung Lie (賴文能) received the B.S., M.S., and Ph.D. degrees in Electrical Engineering, from National Tsing Hua University, Hsinchu, Taiwan, in 1984, 1986, and 1990, respectively. From 1990 to mid 1996, he served as an Assistant Scientist at the Chung Shan Institute of Science and Technology, Taoyuan County, Taiwan, where he worked on the development of infrared imaging systems and target trackers for military applications. In 1996, he joined the Department of Electrical Engineering, National Chung Cheng University, Chia-Yi, Taiwan, where he is currently a Professor and Chair. From 2000 to 2001, he was a Visiting Scholar at the University of Washington, Seattle, WA, USA. His current research interests include image/video compression, networked video transmission, audio/image/video watermarking, multimedia

content analysis, standard-compliant multimedia encryption, infrared image processing, 3-D TV related technologies, 2-D-to-3-D image/video conversion, and industrial inspection using computer vision techniques.