Fall Detection with Unobtrusive Infrared Array Sensors

Xiuyi Fan[1], Huiguo Zhang[2], Cyril Leung[3], and Zhiqi Shen[2]

[1] Swansea University, Swansea, United Kingdom,
[2] Nanyang Technological University, Singapore,
[3] The University of British Columbia, Canada

Abstract. As the world's aging population grows, fall is becoming a major problem in public health. It is one of the most vital risks to the elderly. Many technology based fall detection systems have been developed in recent years with hardware ranging from wearable devices to ambience sensors and video cameras. Several machine learning based fall detection classifiers have been developed to process sensor data with various degrees of success. In this paper, we present a fall detection system using infrared array sensors with several deep learning methods, including *long-short-term-memory* and *gated recurrent unit* models. Evaluated with fall data collected in two different sets of configurations, we show that our approach gives significant improvement over existing works using the same infrared array sensor.

Keywords: fall detection, machine learning, unobtrusive sensing

1 Introduction

Aging is a global challenge faced by many countries in the world. The rapid growth of the aging population puts high demand for relevent assistive technologies supported by various sensor-actuator systems [15]. There are various types of sensors utilized in assisted living, including cameras [24], light sensors, accelerometers [39], temperature sensors, gyroscope, barometer, infrared sensors [32], etc. These sensors are rich data sources for analyzing various aspects of a user's daily life, ranging from health and fitness monitoring, personal biometric signature, navigation and localization [25]. In this context, one particular problem is the detection of falls. Fall is the most vital risk to the elderly's health as over one in every three elderly people suffer from fall consequences [12, 41]. In event of fall, it is urgent to provide immediate treatment of the injured. Thus the quick detection of fall is essential for on time treatment [38].

Technology based fall detection has been of great interest. It has generated a wide range of applied research and has prompted the development of telemonitoring systems to enable the early diagnosis of fall conditions [27]. Mubashir et. al. distinguish fall detection systems into three categories, wearable devices, ambience sensors and cameras [25]. The first category needs the subject of interest wearing a wearable device all the time whereas the last two only to deploy the device in the vicinity of the subject.

In addition to sensor development, different data classification techniques have been developed for fall detection. From raw sensor data, various data processing algorithms have been proposed in the literature. Roughly speaking, there are two schools of methods for fall detection: rule-based methods that detect falls with domain knowledge and machine learning based approaches "learn fall characteristics" from training data [15, 27].

In this work, we present a fall detection system that is based on data collected from Grid-Eye Infrared Array Sensors, which are low cost, low resolution infrared thermal image temperature sensors. These low resolution sensors have less intrusion of privacies when compared with high resolution sensors like RGB cameras. Sensor data is processed with several mainstream deep learning models, including the long short term memory (LSTM) [11] and gated recurrent unit (GRU) models [6]. We have also experimented these models with attention mechanisms as proposed in [7]. We compare our approaches with the fall detection system reported in [22], which also uses the same Grid-Eye sensor, and we show that our approach yields improvement over existing ones.

The rest of this paper is organized as follows. Section 2 introduces several existing works on fall detection. Section 3 introduces deep learning classifiers we developed in this work. Section 4 presents performance evaluation of the developed fall detection system. We conclude the paper and discuss future research directions in section 5.

2 Related Work

Existing fall detection systems can be categorized into three types, wearable devices, camera systems and ambience sensors [25]. Wearable devices are sensors attached to a human body to collect body movements and to recognize activities. Most wearable devices use accelerometers and gyroscopes [16, 4]. In these fall detection systems, sensors are attached to different parts of the user's body such as waist [41], chest [12], and shoes [30]. One major problem with wearable device based methods is that the user has to wear the device all the time, which causes a great amount of inconvenience. Also, users often forgot to wear such devices from time to time.

Camera based fall detection systems normally use RGB cameras [28]. Recently, several studies also use Microsoft Kinect [33, 23]. Camera-based devices are commonly deployed through the elderly's house or at public places. There are two limitations with these systems, privacy intrusion with video monitoring and the lack of system robustness.

Ambience sensor based fall detection systems have also been studied. Different sensors or devices such as doppler radar [19], passive infrared sensors [20, 37, 22, 5], pressure sensors [35, 14], sound sensors [18] and Wi-Fi routers [36] have been tested for fall detection.

Many research has been devoted to the study of fall detection classification algorithms [38, 1]. There are mainly two categories of methods developed, rule-based methods that depend much on domain knowledge and machine learning methods that recognize fall characteristics from sensor data [15, 27]. For instance, [3, 2, 13, 17] are some early fall detection works with threshold-based algorithms. In those works, thresholds are set such that if any of these thresholds is exceeded, then a fall alert is triggered. The major drawback of these approaches is the lack of adaptability and flexibility.

At the same time, various machine learning based fall detection classifiers have been developed [21]. Mainstream machine learning approaches, including decision trees [29], support vector machines (SVM) [34], k-nearest neighbours (k-NN) [8] and hidden Markov models [10] have been applied in fall detection, see e.g., [9, 26, 40, 5]. Many of these approaches rely on manually designed features for classification.

The following works are most relevant to ours. L. Liu et. al. [19] develop a dual Doppler radar system for fall detection. A fusion methodology combines partial decision information from two sensors in three different classifiers, k-NN, SVM and Bayes to form a fall/non-fall decision based on Melfrequency Cepstral Coefficients (MFCC) features. Its performance measured with AUC is 0.88 and 0.97.

Liu et. al. [20] pospose a two-layer hidden Markov model for recognizing a fall event based on the signals of five passive infrared sensors which were placed at different heights on the wall. The associated sensitivity and specificity of the falls algorithm were 92.5% and 93.7%, respectively.

Chen et. al. [5] use 16-by-4 thermopile array sensors for fall detection and elderly tracking. Two sensors are used in their system with a k-NN classifier. They have reached 95.25% sensitivity, 90.75% specificity and 93% accuracy in their experiment. Sixsmith and Johnson [31] developed a *Smart Inactivity Monitor* using array-based detectors which also detects falls.

Mashiyama et. al. [22] propose a system of fall detection using an infrared array sensor. From a data sequence obtained in a fixed window, four manually crafted features, number of consecutive frames, maximum number of pixels, maximum variance of temperature and distance of a maximum temperature pixel, are extracted from the sequence and used to classify falls or non-falls using with the k-NN algorithm. Experiment results with their testing data show that their system reaches 94% accuracy.

3 Fall Detection Classifiers

At the core of our fall detection system is the infrared array sensor, Grid-Eye (AMG8832). A Grid-Eye sensor outputs an 8-pixel by 8-pixel temperature distribution in its 60-degree field of view at a maximum 10-frame per second rate. Its maximum detection distance is 5m if there is a $\geq 4°$C temperature difference between the foreground object and the background ambience. We use a ZigBee CC2530 as a microprocessor to control the sensor via an I^2C bus as shown in Figure 1. The measured temperature distribution is sent to another ZigBee CC2530 at a 10Hz rate. A standard PC is then used for data processing and classification.

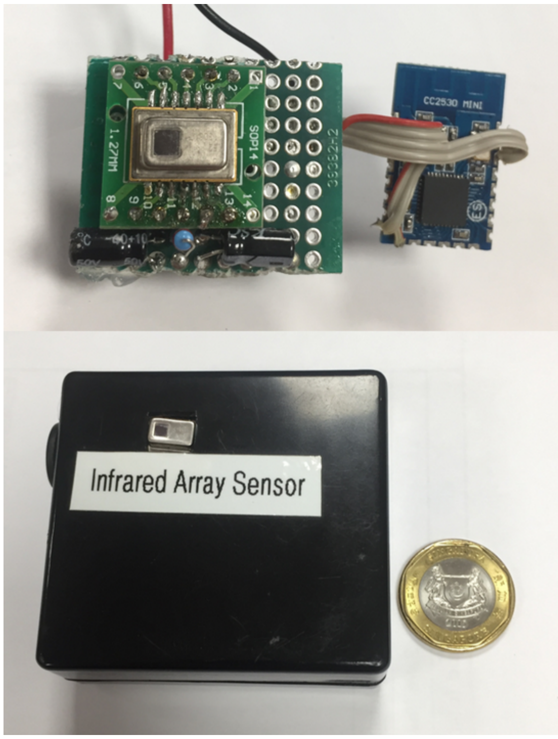

Fig. 1. The Grid-Eye sensor package used in our experiment.

Although a Grid-Eye sensor measures temperature in a large range (-20°C to 100°C), its temperature accuracy is only 3.0°C. Since thermal image based fall detection depends on correctly identifying the abrupt movement of a human body, the ability to recognize the subtle temperature difference between the human body and the ambience is the key to ensure correct detections. However, as illustrated in Figure 2, data obtained from Grid-Eye sensors is noisy. (In this figure, warm colour indicates high temperature.) Thus, we develop a fall detection system with two main components: (1) data filters for pre-processing and (2) neural networks for classification. As illustrated in Figure 3, data produced by the Grid-Eye is firstly filtered with one of the filters. Filtered data is then passed to neural network classifiers.

Three filters, Median, Gaussian and Wavelet, have been experimented in this work. For neural network classifiers, we have experimented with two-layer

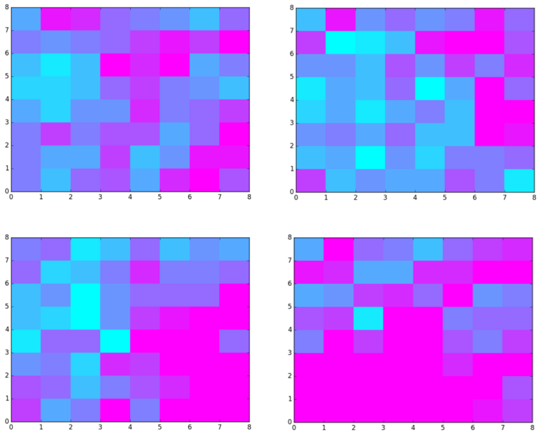

Fig. 2. Illustration of Grid-Eye images. Top left: no person in Grid-Eye's field of view. Top right: a person standing on the right-hand side. Bottom left: a person falling from the right-hand side. Bottom right: a person lying in front of the Grid-Eye.

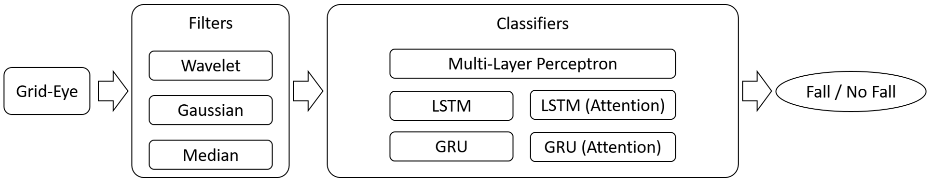

Fig. 3. Fall Detection Classification Workflow.

perceptron networks (Figure 4), long short-term memory (LSTM) networks and gated recurrent unit (GRU) networks (Figure 5), each with and without attention links.

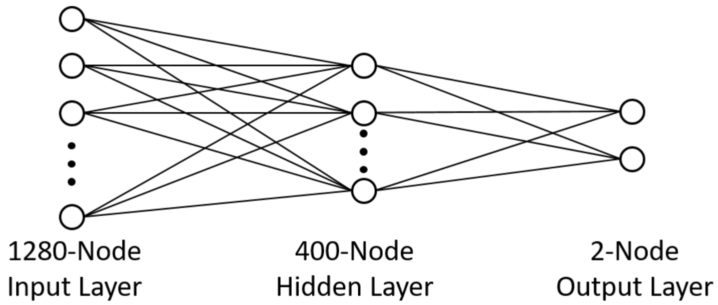

Fig. 4. Two-layer Fully Connected Perceptron Network.

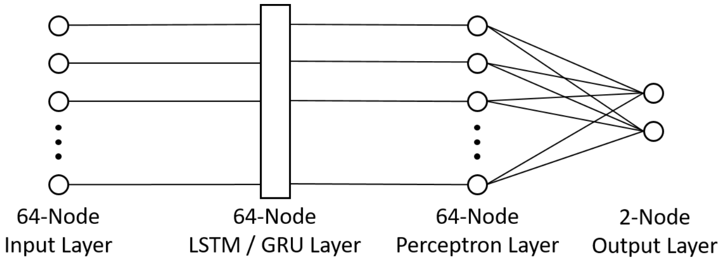

Fig. 5. LSTM / GRU Networks.

As illustrated in Figure 6, the developed system works as follows. At each time step t, the Grid-Eye outputs thermal reading represented with a 1×64 vector. To detect fall, we examine data collected in a 2-second (outer) window. Since the Grid-Eye is running at 10Hz, 20 1×64 vectors are collected during each (outer) window. We then filter data stored in this outer window with one of the three filters. For both median and Gaussian filters, an inner window of size 5 is used. For the wavelet filter, we use Daubechies 4 tap wavelet. The filtering process does not change the size of the data. Filtered data is then sent to neural networks for classification.

Two-layer perceptron networks with the following configuration are selected for their simplicity. The input layer contains $64 \times 20 = 1280$ nodes (64 is the length of the Grid-Eye output vector and 20 is the size of the outer window).

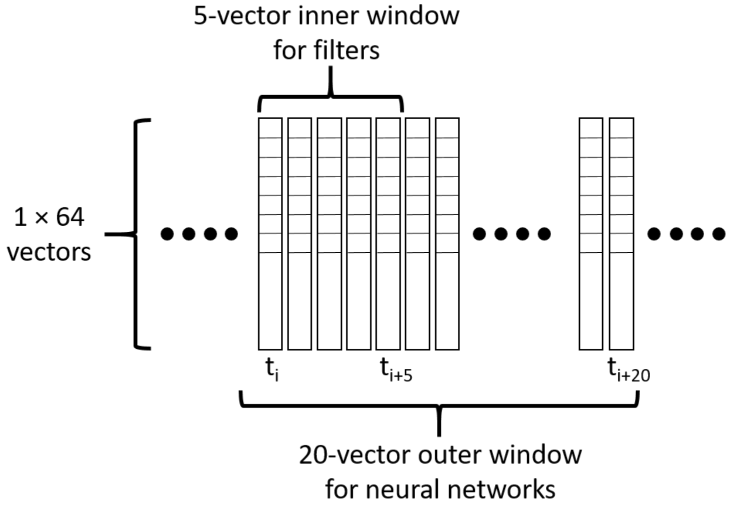

Fig. 6. Data layout for filters and classifiers.

The fully connected hidden layer contains 400 nodes. The output layer contains 2 nodes (indicating a fall and not a fall, respectively).

LSTM and GRU networks have seen many successes in recent years. They both contain "memory structures", i.e., LSTM cells and GRU units, to store past information. As illustrated in Figure 5, the input layers of our LSTM and GRU networks both contain 64 nodes. There is a fully connected perceptron layer with 64 nodes between the LSTM / GRU layer and the 2-node output layer. The LSTM model can be described with the following equations.

$$i = \sigma(x_t U^i + s_{t-1} W^i) \tag{1}$$

$$f = \sigma(x_t U^f + s_{t-1} W^f) \tag{2}$$

$$o = \sigma(x_t U^o + s_{t-1} W^o) \tag{3}$$

$$g = tanh(x_t U^g + s_{t-1} W^g) \tag{4}$$

$$c_t = c_{t-1} \circ f + g \circ i \tag{5}$$

$$s_t = tanh(c_t) \circ o \tag{6}$$

Here, σ is the sigmoid function. $\circ$ denotes element-wise multiplication. x_t is the input at time t. s_t is the output of the cell at time t. Us and Ws are weight matrices connecting various components. Specifically, in our system, x_t is a 1-by-64 vector; s_t is a 1-by-64 vector; Us, are 64-by-64 matrices; Ws are 64-by-64 matrices.

GRU [6] is a recently proposed variation of the LSTM model. The main difference is that, instead of using three gates to control memory updates, a

GRU unit uses only two gates. Formally, a GRU model can be described with the following equations:

$$z = \sigma(x_t U^z + s_{t-1} W^z) \tag{7}$$

$$r = \sigma(x_t U^r + s_{t-1} W^r) \tag{8}$$

$$h = tanh(x_t U^h + (s_{t-1} \circ r) W^h) \tag{9}$$

$$s_t = (1 - z) \circ h + z \circ s_{t-1} \tag{10}$$

Again, σ is the sigmoid function. x_t is the input at time t. h is the output. s_t is the internal state of a GRU unit at time t. The size of Us and Ws are the same as in LSTM. Essentially, we use the same network structure as our LSTM implementation, with LSTM cells replaced by GRU units.

Introducing attention mechanism into both LSTM and GRU models in this work is very simple. Conceptually, the attention mechanism provides a means for specifying the relative importance of each frame in a classification window (20-frames in our case). For instance, s_t in Equation 6 for $t = 20$ not only depends on s_{19} but also (directly) depends on all previous s_i, for $1 \leq i \leq 19$, i.e.,

$$s_{20} = \sum_{0 \leq i < 20} \omega_i s_i, \tag{11}$$

for some ω_i also learned with backward propagation though time as U and W.

4 Performance Evaluation

To evaluate the performance of the developed system, we conduct fall detection experiments in our laboratory environment (Figure 7). In our test, we have created a dataset with 312 falls in two sets of configurations. As illustrated in Figure 8, in the first set of experiments, the testing subject falls perpendicular to the Grid-Eye sensor at A, B and C three different positions. In the second set of experiments, the testing subject falls parallel to the Grid-Eye sensor, also at A, B and C three different positions. In both configurations, negative examples including randomly walking in the room, slowly sitting down, jumping, running and laying down in front of the sensor have been performed. The dataset has been created in multiple sessions crossing several days with ambient temperature ranging from 19°C to 23°C.

For evaluation, we have divided the dataset into a training set with 240 falls and a testing set with 72 falls with each falling position contains exactly the same number of falls. Since robust fall detection requires high ratings in both precision and recall, reducing both false positives and false negatives, we compare results

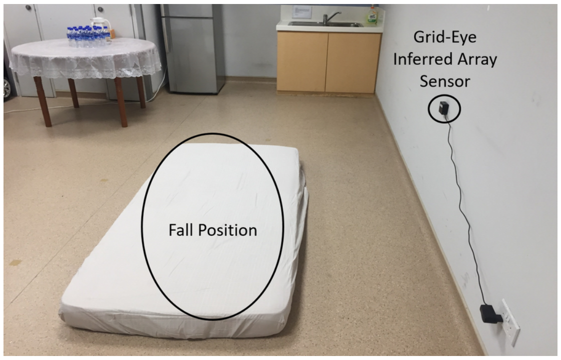

Fig. 7. Testing Environment (illustrated for one testing configuration).

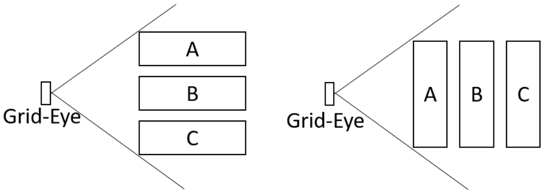

Fig. 8. Illustration of Experiment Configurations. In configurations shown on the left, the testing subject falls in directions perpendicular to the Grid-Eye at positions A, B and C. In configurations shown on the right, the testing subject falls in directions parallel to the Grid-Eye at positions A, B and C.

with F1 scores for each test case, defined as follows.

$$\begin{aligned}\text{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}},\\ \text{Recall} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}},\\ \text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.\end{aligned}$$

Table 1: Experiment Results from the MLP classifier.

	F-Score	Precision	Recall	Total	True Positive	False Negative
No Filter (H)	0.972	0.972	0.972	36	35	1
No Filter (V)	0.679	0.522	0.972	67	35	1
Median Filter (H)	0.986	0.972	1	37	36	0
Median Filter (V)	0.666	0.619	0.722	42	26	10
Gaussian Filter (H)	0.972	0.972	0.972	36	35	1
Gaussian Filter (V)	0.693	0.666	0.722	39	26	10
Wavelet Filter (H)	0.972	0.947	1	38	36	0
Wavelet Filter (V)	0.658	0.568	0.75	46	27	9

Table 2: Experiment Results from the LSTM classifier.

	F-Score	Precision	Recall	Total	True Positive	False Negative
No Filter (H)	0.956	1	0.916	33	33	3
No Filter (V)	0.864	0.777	0.972	45	35	1
Median Filter (H)	1	1	1	36	36	0
Median Filter (V)	0.805	0.805	0.805	36	29	7
Gaussian Filter (H)	0.986	0.972	1	37	36	0
Gaussian Filter (V)	0.805	0.805	0.805	36	29	7
Wavelet Filter (H)	0.986	0.972	1	37	36	0
Wavelet Filter (V)	0.746	0.659	0.861	47	31	5

Experiment results from our systems are shown in Table 1-5. In each table, rows labelled with (H) and (V) are experment results from falls parallel and perpendicular to the Grid-Eye sensors, respectively. Overall, we make the following observations.

- Measured by F1 scores, all classifiers perform better in settings where users fall parallelly to the sensor. This indicates that falling-parallel-to-the-sensor is intrinsically easier to classify than falling-perpendicular-to-the-sensor.
- Introducing filters specifically to remove noise improves the performance in certain cases. Amongst three filters tested, the simple median filter performs better than the other two.

Table 3: Experiment Results from the LSTM-ATT classifier.

	F-Score	Precision	Recall	Total	True Positive	False Negative
No Filter (H)	0.972	0.947	1	38	36	0
No Filter (V)	0.857	0.804	0.916	41	33	3
Median Filter (H)	0.947	0.9	1	40	36	0
Median Filter (V)	0.819	0.723	0.944	47	34	2
Gaussian Filter (H)	0.96	0.923	1	39	36	0
Gaussian Filter (V)	0.735	0.627	0.888	51	32	4
Wavelet Filter (H)	0.944	0.944	0.944	36	34	2
Wavelet Filter (V)	0.749	0.681	0.833	44	30	6

Table 4: Experiment Results from the GRU classifier.

	F-Score	Precision	Recall	Total	True Positive	False Negative
No Filter (H)	0.972	0.9447	1	38	36	0
No Filter (V)	0.825	0.75	0.916	44	33	3
Median Filter (H)	0.935	0.878	1	41	36	0
Median Filter (V)	0.819	0.723	0.944	47	34	2
Gaussian Filter (H)	0.972	0.972	0.972	36	35	1
Gaussian Filter (V)	0.722	0.638	0.833	47	30	6
Wavelet Filter (H)	0.911	0.837	1	43	36	0
Wavelet Filter (V)	0.692	0.642	0.75	42	27	9

Table 5: Experiment Results from the GRU-ATT classifier.

	F-Score	Precision	Recall	Total	True Positive	False Negative
No Filter (H)	0.935	0.878	1	41	36	0
No Filter (V)	0.904	0.891	0.916	37	33	3
Median Filter (H)	0.986	0.972	1	37	36	0
Median Filter (V)	0.742	0.764	0.722	34	26	10
Gaussian Filter (H)	0.945	0.921	0.972	38	35	1
Gaussian Filter (V)	0.739	0.729	0.75	37	27	9
Wavelet Filter (H)	0.933	0.897	0.972	39	35	1
Wavelet Filter (V)	0.722	0.638	0.833	47	30	6

- There is no clear winner between LSTM models and GRU models. The memory ability of both models works well.
- Introducing attention mechanisms in both LSTM and GRU models does not consistently improve the performance. This may suggest that fall detection takes information from all frames containing a fall equally and it gives no advantage to focus the detection at any single moment of the fall.
- When the classification problem is easy (parallel settings), MLP does not expose its weakness; however, when the problem gets more difficult (perpendicular settings), models explicitly recording previous information perform significantly better.

In order to put our results into perspective, we compare our approaches with the model presented in [22], which uses the same Grid-Eye sensor with a k-NN classifier with four manually crafted features. We replicate their system and tested on our dataset, the comparison results are shown in Table 6 (perpendicular to the sensor) and 7 (parallel to the sensor). From these two tables, we see that their approach also performs better when falls are parallel to the sensor. However, overall, their k-NN classifier with manually crafted features performs worse than any of our neural network based approaches with data filtering.

Table 6: Fall Detection Performance (Falls are perpendicular to the Grid-Eye).

	Precision	Recall	F1
GRU-ATT	0.891	0.916	0.904
GRU	0.75	0.916	0.825
LSTM-ATT	0.804	0.916	0.857
LSTM	0.777	0.972	0.864
MLP	0.666	0.722	0.693
k-NN [22]	0.52	1	0.68

Table 7: Fall Detection Performance (Falls are parallel to the Grid-Eye).

	Precision	Recall	F1
GRU-ATT	0.97	1	0.99
GRU	0.972	0.972	0.972
LSTM-ATT	0.947	1	0.972
LSTM	1	1	1
MLP	0.972	1	0.986
k-NN [22]	0.83	0.97	0.9

We have also experimented with different outer window size for the fall detection using four different classifiers. In the original setting, the outer window is 20 (See Figure 6), meaning that each fall detection occurs in a 2-second window, as the Grid-Eye is running at 10Hz. In Table 8 and 9, we show fall detection result with outer window being 30, we see that the performances are considerably lower for all four classifiers (the Median filter has been used in these experiments). We interpret these results as: since fall is an instantaneous event, increasing the window size does not help improving the detection performance.

Table 8: Fall Detection Performance with 3-seconds detection window (Falls are perpendicular to the Grid-Eye).

	Precision	Recall	F1
GRU-ATT	0.632	0.861	0.729
GRU	0.731	0.833	0.779
LSTM-ATT	0.695	0.888	0.780
LSTM	0.82	0.888	0.853

Table 9: Fall Detection Performance with 3-seconds detection window (Falls are parallel to the Grid-Eye).

	Precision	Recall	F1
GRU-ATT	0.7	0.972	0.813
GRU	0.809	0.944	0.871
LSTM-ATT	0.875	0.972	0.921
LSTM	0.947	1	0.972

5 Conclusion

Fall is a major health threat to the elderly. In event of fall, it is urgent to provide immediate treatment to the injured people. In this paper, we present a fall detection system using Grid-Eye infrared array sensor. Due to its low spatial resolution, infrared array sensor incurs little privacy intrusion and can be deployed to sensitive areas such as washrooms, which are known to be fall-prone. For data processing, we have taken a two-step approach: (1) pre-processing data filtering and (2) machine learning classification with neural networks. For filtering, we have experimented with Wavelet, Gaussian and Median filters. For classification, we have experimented with several deep learning models, including multi-layer perceptrons, LSTM and GRU. To evaluate our approaches, we have created a dataset containing over 300 falls in multiple configurations. We then compare our work with an existing work using the same infrared array sensor but with different classification techniques and show significantly improved classification accuracy. In the future, we would like to (1) perform in depth theoretical study, including computational complexity analysis, of the proposed methods, (2) deploy our system to nursing homes for real-world experiment and (3) explore fall detection with other ambience sensor systems and deployment configurations.

References

1. F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS ONE*, 7(5), 2012.
2. A. K. Bourke and G. M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med Eng Phys*, 30(1):84–90, Jan 2008.
3. A. K. Bourke, J. V. O'Brien, and G. M. Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture*, 26(2):194–199, Jul 2007.
4. J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. Wearable sensors for reliable fall detection. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 3551–3554, Jan 2005.
5. Wei-Han Chen and Hsi-Pin Ma. A fall detection system based on infrared array sensors with tracking capability for the elderly at home. In *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, pages 428–434, Oct 2015.
6. Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.
7. Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *CoRR*, abs/1506.07503, 2015.
8. Sahibsingh A. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Trans. Systems, Man, and Cybernetics*, 6(4):325–327, 1976.
9. Raghu K. Ganti, Praveen Jayachandran, Tarek F. Abdelzaher, and John A. Stankovic. Satire: A software architecture for smart attire. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, MobiSys '06, pages 110–123, New York, NY, USA, 2006. ACM.
10. Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *IJPRAI*, 15(1):9–42, 2001.
11. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
12. J. Y. Hwang, J. M. Kang, Y. W. Jang, and H. C. Kim. Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 1, pages 2204–2207, Sept 2004.
13. M. Kangas, A. Konttila, I. Winblad, and T. Jamsa. Determination of simple thresholds for accelerometry-based parameters for fall detection. *Conf Proc IEEE Eng Med Biol Soc*, 2007:1367–1370, 2007.
14. Lars Klack, Christian Möllering, Martina Ziefle, and Thomas Schmitz-Rode. *Future Care Floor: A Sensitive Floor for Movement Monitoring and Fall Detection in Home Environments*, pages 211–218. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
15. Simon Kozina, Hristijan Gjoreski, Matjaž Gams, and Mitja Luštrek. *Efficient Activity Recognition and Fall Detection Using Accelerometers*, pages 13–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

16. Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pages 138–143, June 2009.
17. Qiang Li, Gang Zhou, and John A. Stankovic. Accurate, fast fall detection using posture and context information. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 443–444, New York, NY, USA, 2008.
18. Y. Li, Z. Zeng, M. Popescu, and K. C. Ho. Acoustic fall detection using a circular microphone array. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 2242–2245, Aug 2010.
19. L. Liu, M. Popescu, M. Skubic, and M. Rantz. An automatic fall detection framework using data fusion of doppler radar and motion sensor network. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5940–5943, Aug 2014.
20. Tong Liu, Xuemei Guo, and Guoli Wang. Elderly-falling detection using distributed direction-sensitive pyroelectric infrared sensor arrays. *Multidimensional Systems and Signal Processing*, 23(4):451–467, 2012.
21. Mitja Lustrek and Bostjan Kaluza. Fall detection and activity recognition with machine learning. *Informatica (Slovenia)*, 33:197–204, 2009.
22. S. Mashiyama, J. Hong, and T. Ohtsuki. A fall detection system using low resolution infrared array sensor. In *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pages 2109–2113, Sept 2014.
23. Georgios Mastorakis and Dimitrios Makris. Fall detection system using kinect's infrared sensor. *Journal of Real-Time Image Processing*, 9(4):635–646, 2014.
24. S. G. Miaou, Pei-Hsu Sung, and Chia-Yuan Huang. A customized human fall detection system using omni-camera images and personal information. In *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, 2006. D2H2.*, pages 39–42, April 2006.
25. Muhammad Mubashir, Ling Shao, and Luke Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013.
26. H. Nait-Charif and S. J. McKenna. Activity summarisation and fall detection in a supportive home environment. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 323–326 Vol.4, Aug 2004.
27. N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy. Fall detection - principles and methods. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1663–1666, Aug 2007.
28. C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 2, pages 875–880, May 2007.
29. S. Rasoul Safavian and David A. Landgrebe. A survey of decision tree classifier methodology. *IEEE Trans. Systems, Man, and Cybernetics*, 21(3):660–674, 1991.
30. S. Y. Sim, H. S. Jeon, G. S. Chung, S. K. Kim, S. J. Kwon, W. K. Lee, and K. S. Park. Fall detection algorithm for the elderly using acceleration sensors on the shoes. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4935–4938, Aug 2011.

31. A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing*, 3(2):42–47, April 2004.
32. Andrew Sixsmith and Neil Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing*, 3:42–47, 2004.
33. E. E. Stone and M. Skubic. Fall detection in homes of older adults using the microsoft kinect. *IEEE Journal of Biomedical and Health Informatics*, 19(1):290–301, Jan 2015.
34. Johan A. K. Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.
35. Huan-Wen Tzeng, Mei-Yung Chen, and J. Y. Chen. Design of fall detection system with floor pressure and infrared image. In *2010 International Conference on System Science and Engineering*, pages 131–135, July 2010.
36. H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li. Rt-fall: A real-time and contactless fall detection system with commodity wifi devices. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2016.
37. Piotr Wojtczuk, David Binnie, Alistair Armitage, Tim Chamberlain, and Carsten Giebeler. A touchless passive infrared gesture sensor. In *Proceedings of the Adjunct Publication of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13 Adjunct, pages 67–68, New York, NY, USA, 2013. ACM.
38. Xinguo Yu. Approaches and principles of fall detection for elderly and patient. In *HealthCom 2008 - 10th International Conference on e-health Networking, Applications and Services*, pages 42–47, July 2008.
39. Tong Zhang, Jue Wang, Ping Liu, and Jing Hou. Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. In *IJCSNS International Journal of Computer Science and Network Security*, 2006.
40. Tong Zhang, Jue Wang, Liang Xu, and Ping Liu. *Fall Detection by Wearable Sensor and One-Class SVM Algorithm*, pages 858–863. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
41. J. Zheng, G. Zhang, and T. Wu. Design of automatic fall detector for elderly based on triaxial accelerometer. In *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pages 1–4, June 2009.