ISSN 2007-9737

Classification of Fall Events in the Elderly Using a Thermal Sensor and Machine Learning Techniques

Arnoldo Díaz-Ramírez, Julia Diaz-Escobar*, Verónica Quintero-Rosas,
Rosendo Moncada-Sánchez

Tecnológico Nacional de México / Instituto Tecnológico de Mexicali,
Departamento de Sistemas y Computacion, Mexicali,
Mexico

{adiaz, jdiaz, veronicaquintero}@itmexicali.edu.mx, rosendo.msa@gmail.com

Abstract. As reported by the World Health Organization, falls constitute the second leading cause of unintentional injury death worldwide. Particularly, adults older than 60 years suffer the most significant number of fatal falls or serious injuries, with nearly 30% of individuals over 65 reporting at least one fall annually, a risk that increases with age. The anticipated growth in life expectancy and the resulting larger aging population accentuates the economic burden associated with falls. Consequently, identifying effective strategies for fall prevention and early detection in the elderly has become relevant. This study proposes a non-invasive fall detection system based on a thermal sensor and a supervised machine-learning algorithm. The experimental dataset, generated by students through simulations of both fall and non-fall events, included the recording of room temperatures using a thermal sensor, along with the associated data labeling. For fall event detection, we evaluated three well-known supervised machine learning models: a Support Vector Machine, a Random Forest, and a Convolutional Neural Network. The experimental results demonstrate that these models exhibit robust capabilities in distinguishing between falls and non-fall events, consistently achieving performances above 95% across various evaluation metrics.

Keywords. Elderly care, machine learning, sensor monitoring, fall events.

1 Introduction

According to the World Health Organization (WHO) [29], falls constitute the second leading cause of unintentional injury death worldwide.

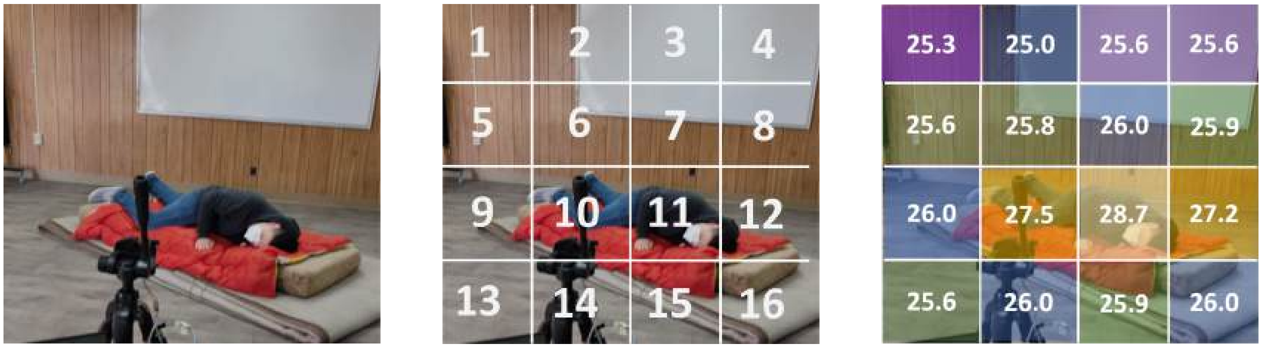
Each year, 37.3 million falls require medical attention, and an estimated 684,000 become fatal. Notably, adults older than 60 years suffer the most significant number of fatal falls or serious injuries. Almost 30% of adults over 65 years report at least one fall yearly [1], increasing the risk with age [29].

The most common causes of falls in elderly individuals are environment-related factors and disorders related to gait, balance, or weakness [24]. Additionally, older individuals with mobility impairments, cognitive deficits, chronic conditions, geriatric syndromes, and the use of particular medications are at an increased risk of experiencing falls [12].

From a financial perspective, elderly falls impact the economic concerns of government-funded programs. Several studies estimated annual costs of billions of dollars in expenditures for medical fall treatments [6]. Moreover, the economic burden would be expected to grow due to the rising life expectancy, leading to a larger aging population.

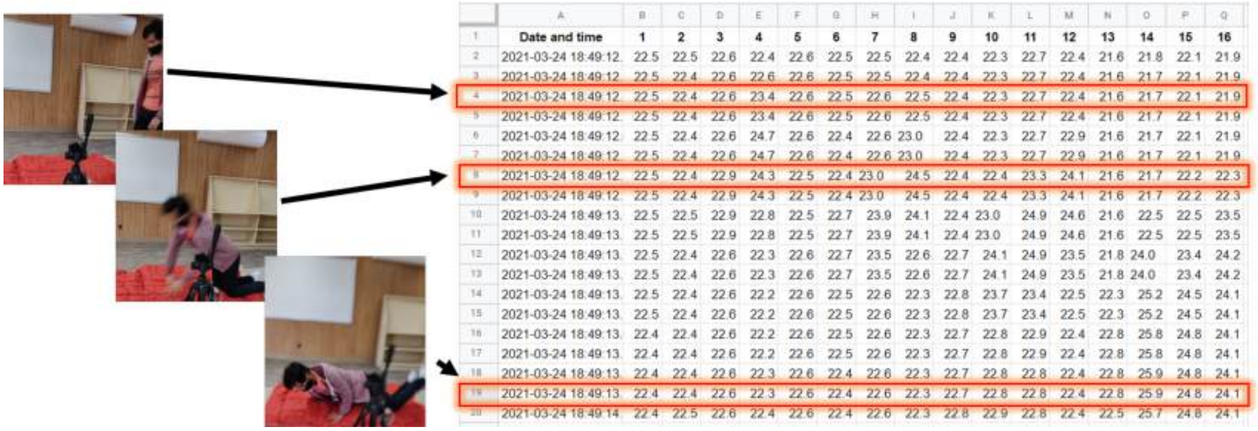
As reported by the WHO, the population of individuals older than 60 will double (2.1 billion) by 2050, which is 22% of the global population [28]. Therefore, identifying strategies for fall prevention and early detection in elderly individuals becomes a topic of relevance. Over the past decades, fall prevention and detection have been active research areas [23]. Several strategies, including risk factors reduction, exercise routines, environmental modifications, and education programs, have demonstrated effectiveness in preventing falls [24]. However, while fall prevention can reduce the occurrence of falls, it does not eliminate the possibility of a fall event.

doi: 10.13053/CyS-28-4-4809



(a) (b) (c)

Fig. 1. Frame temperature readings: (a) Frame at time t. (b) 4×4 thermal sensor resolution. (c) Room temperatures captured by the sensor



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	**Date and time**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
2	2021-03-24 18:49:12	22.5	22.5	22.6	22.4	22.6	22.5	22.5	22.4	22.4	22.3	22.7	22.4	21.6	21.8	22.1	21.9
3	2021-03-24 18:49:12	22.5	22.4	22.6	22.6	22.6	22.5	22.5	22.4	22.4	22.3	22.7	22.4	21.6	21.7	22.1	21.9
4	2021-03-24 18:49:12	22.5	22.4	22.6	23.4	22.6	22.5	22.6	22.5	22.4	22.3	22.7	22.4	21.6	21.7	22.1	21.9
5	2021-03-24 18:49:12	22.5	22.4	22.6	23.4	22.6	22.5	22.6	22.5	22.4	22.3	22.7	22.4	21.6	21.7	22.1	21.9
6	2021-03-24 18:49:12	22.5	22.4	22.6	24.7	22.6	22.4	22.6	23.0	22.4	22.3	22.7	22.9	21.6	21.7	22.1	21.9
7	2021-03-24 18:49:12	22.5	22.4	22.6	24.7	22.6	22.4	22.6	23.0	22.4	22.3	22.7	22.9	21.6	21.7	22.1	21.9
8	2021-03-24 18:49:12	22.5	22.4	22.9	24.3	22.5	22.4	23.0	24.5	22.4	22.4	23.3	24.1	21.6	21.7	22.2	22.3
9	2021-03-24 18:49:12	22.5	22.4	22.9	24.3	22.5	22.4	23.0	24.5	22.4	22.4	23.3	24.1	21.6	21.7	22.2	22.3
10	2021-03-24 18:49:13	22.5	22.5	22.9	22.8	22.5	22.7	23.9	24.1	22.4	23.0	24.9	24.6	21.6	22.5	22.5	23.5
11	2021-03-24 18:49:13	22.5	22.5	22.9	22.8	22.5	22.7	23.9	24.1	22.4	23.0	24.9	24.6	21.6	22.5	22.5	23.5
12	2021-03-24 18:49:13	22.5	22.4	22.6	22.3	22.6	22.7	23.5	22.6	22.7	24.1	24.9	23.5	21.8	24.0	23.4	24.2
13	2021-03-24 18:49:13	22.5	22.4	22.6	22.3	22.6	22.7	23.5	22.6	22.7	24.1	24.9	23.5	21.8	24.0	23.4	24.2
14	2021-03-24 18:49:13	22.5	22.4	22.6	22.2	22.6	22.5	22.6	22.3	22.8	23.7	23.4	22.5	22.3	25.2	24.5	24.1
15	2021-03-24 18:49:13	22.5	22.4	22.6	22.2	22.6	22.5	22.6	22.3	22.8	23.7	23.4	22.5	22.3	25.2	24.5	24.1
16	2021-03-24 18:49:13	22.4	22.4	22.6	22.2	22.6	22.5	22.6	22.3	22.7	22.8	22.9	22.4	22.8	25.8	24.8	24.1
17	2021-03-24 18:49:13	22.4	22.4	22.6	22.2	22.6	22.5	22.6	22.3	22.7	22.8	22.9	22.4	22.8	25.8	24.8	24.1
18	2021-03-24 18:49:13	22.4	22.4	22.6	22.3	22.6	22.4	22.6	22.3	22.7	22.8	22.8	22.4	22.8	25.9	24.8	24.1
19	2021-03-24 18:49:13	22.4	22.4	22.6	22.3	22.6	22.4	22.6	22.3	22.7	22.8	22.8	22.4	22.8	25.9	24.8	24.1
20	2021-03-24 18:49:14	22.4	22.5	22.6	22.4	22.6	22.4	22.6	22.3	22.8	22.9	22.8	22.4	22.5	25.7	24.8	24.1

Fig. 2. Final CSV file composed by 25 frames (rows) and 16 temperatures (cols)

Conversely, fall detection techniques focus on recognizing falls and alerting when a fall event has occurred [27]. In this work, we introduced a fall detection model based on non-wearable devices and machine learning techniques.

The main contributions of this study are, firstly, the establishment of a dataset containing room temperature values through the utilization of a thermal sensor and, secondly, the application of machine learning techniques for the classification of fall and non-fall events.

2 Related Work

Fall detector methods can be broadly categorized into wearable and non-wearable device-based approaches. Wearable devices rely on clothing embedded with sensors, including accelerometers, gyroscopes, electro-myography, and pressure sensors, to discern the subject's motion and location [19, 23].

Accelerometers, in particular, have been widely used for fall detection in wearable systems [19]. However, wearable-based systems may not be a good choice for older adults. Wearable devices require subjects to wear the sensors actively and, in some cases, need to be constantly charging (e.g., smartwatches, smartphones).

Moreover, wearable devices may be uncomfortable, easily misplaced, or forgotten by elderly individuals. Unlike wearable devices, non-wearable devices are less invasive.

Table 1. Hyper-parameter values for each ML model

Hyper-parameter	Selection
CNN	
layers	1, 2,3
nodes (1st layer)	[32:256, step=32]
nodes (2nd layer)	[32:256, step=32]
nodes (3rd layer)	[32:256, step=32]
pooling	average, maximum
fully connected	[16:256, step=16]
activation function	ReLU, tanh
optimizer	SGD, Adam
learning rate	[10^{-5}:10^{-2}, step=×10]
batch size	1, 16, 32, 64
SVM	
C-value	[10^{-2}:10^{3}, step=×10]
gamma	[10^{-4}:1, step=×10]
kernel	Linear, polynomial, radial, sigmoid
RF	
n_estimators	30, 50, 100, 300
max_depth	None, 3, 10, 30, 50, 100
min_samples_leaf	3, 5, 10, 30
max_features	None, auto, sqrt, log 2
bootstrap	True, False

Non-wearable devices can be further divided into ambiance and vision-based sensors.

Sound, temperature, visual, and vibrational sensors, among others, fall into the category of non-wearable devices [19, 23]. Díaz-Ramírez et al. [5] introduced a wireless sensor network (WSN)-based fall detection system that relies on sound analysis. In this system, nodes detect falls by analyzing captured acoustic signals.

The model employs a signal-processing algorithm utilizing cross-correlation to measure the similarity between the sampled signal and a reference template signal characterizing a fall event.

If these signals exhibit similarity, the Mel-frequency cepstral coefficients (MFCC) of the fall sound are then extracted. Subsequently, pattern recognition is performed using the dynamic time warping (DTW) method. The system demonstrated a detection rate of 90% in the absence of acoustic interference and 83% in the presence of TV noise.

Another interesting work is proposed by Nishio et al. [20], where they present a fall detection model using a single Microwave Doppler sensor and applying the Hidden Markov Model (HMM) in continuous wave Doppler mode. The Microwave Doppler sensor is mounted on the ceiling, emitting microwaves in a downward direction.

When any activity occurs within the microwave range, the resulting output signal contains information about the activity, with a frequency proportional to the activity's velocity. Fall and non-fall detection models are created by aggregating activities that yield high likelihoods. The proposed HMM model achieved an accuracy of 95%.

Visual-based approaches also have been explored for fall detection [31]. Mecocci et al. [18] presented a method for automatic fall detection utilizing a Microsoft Kinect sensor, with a focus on processing depth data exclusively. Predefined rules from temporary-sequence data analysis carried out fall detection. The model obtained sensitivity and specificity of 62.4-80.3% and 92.5-97.7%, respectively.

Hung et al. [9] introduced a 3D-based approach for fall detection using multiple RGB cameras. The authors utilized predefined thresholds of the measures of humans' heights and occupied areas to distinguish fallings. The visual-based model achieved sensitivity and specificity rates, ranging from 88% to 95.8% and 96% to 100%, respectively.

Related to our work, Mashiyama et al. [17] presented a system designed to detect fall events utilizing an 8×8 infrared array sensor for room temperature analysis. The detection process involves employing a k-nearest neighbor (k-NN) algorithm. The model demonstrated a commendable accuracy rate of 95.8%.

Taniguchi et al. [26] proposed a fall detection system using two 16×16 thermal sensors attached to the ceiling and the wall of the subject room. The authors' system detects different posture transitions using predefined thresholds derived from training data. The model exhibited a

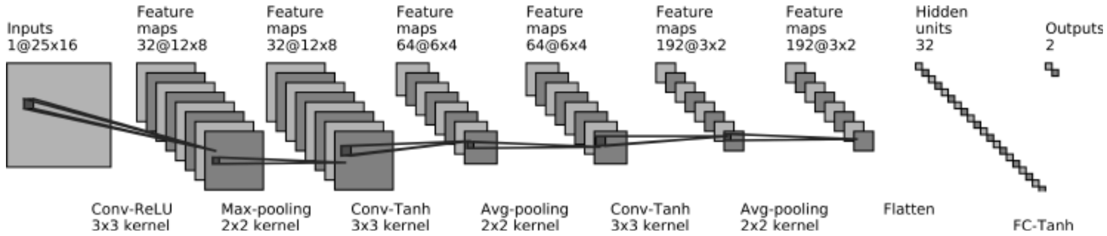


Fig. 3. Resulting CNN architecture after hyperparameter tuning

notable accuracy rate, achieving 95.5% accuracy in fall detection.

3 Materials and Methods

3.1 Machine Learning Techniques

For the fall and non-fall event classification, three supervised machine learning models were evaluated: a Support Vector Machine (SVM) [4], a Random Forest (RF) [8], and a Convolutional Neural Network (CNN) [14]. These models have shown outstanding performance for several machine learning tasks [3, 25, 16].

The SVM model [4] is one of the most known classifiers due to its solid mathematical foundation. Given a set of pairs $\{(\mathbf{x}_i,\ y_i)|\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{Z},\ i = 1,\ 2,\ \ldots,\ m\}$, where y_i indicates the class where the real vector $\mathbf{x}_i$ belongs, the SVM performs classification by constructing a hyperplane in a higher dimensional space that distinguishes one class from the others.

The hyperplane can be expressed as $\mathbf{w}^T\mathbf{x}_i - b = 0$, where $\mathbf{w}^T$ is the transpose of the normal vector to the hyperplane and b is a constant. For the two-class problem, the hyper-plane, or decision boundary, is constructed by solving the following optimization problem:

$$\text{minimize } ||\mathbf{w}||, \tag{1}$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1. \tag{2}$$

On the other hand, the RF classifier [8] is characterized by its simplicity, ease of comprehension, resistance to overfitting, and interpretability of results. The RF algorithm is based on a set of Decision Trees (DTs). Each DT is constructed by randomly selecting data from the training set, employing a technique called bagging [2]. The models generated from these data samples are trained independently, and the algorithm makes its classification decision based on the majority vote of the DTs.

Table 2. ML results

Model	ACC	BACC	AUC-ROC
Mashiyama	0.958	–	–
Taniguchi	0.955	–	–
CNN	0.96	0.95	0.99
SVM	0.99	0.98	0.99
RF	0.99	0.99	0.99

In recent years, CNNs [14] have gained significant popularity. CNNs primarily rely on convolutional layers, where the pixel matrix (or the output matrix from the preceding layer) undergoes convolution with various filters to extract distinctive feature maps. These filters consist of multiple weights that are updated during the network training process. Alongside convolutional layers, CNNs incorporate pooling layers, which employ global, average, or maximum operations to reduce the height and width dimensions of the feature maps.

Similar to conventional artificial neural networks, convolutional networks also include activation functions (such as ReLu, Tanh, sigmoid, etc.) and fully connected layers, commonly referred to as dense layers. The versatility of CNNs in capturing hierarchical and spatial features has contributed to their widespread adoption in various applications [7].

3.2 Dataset

For the dataset, simulations of both fall and non-fall events were conducted by students. Data was collected using a thermal sensor (Omron D6T-44L-06) connected to an Arduino microcontroller. The sensor transmitted sets of bytes, which were converted into integer values representing the room temperatures. These temperatures are stored in CSV files through a Python script. The thermal sensor has a frame resolution of 4×4 and covers a detection area of 2.5×2.5 meters at a distance of three meters.

In each frame, 16 temperature values were captured and recorded at a specific time t (refer to Figure 1). These 16 frame temperatures are organized as a row in a CSV file, resulting in a total of 25 frames (rows) per file, as illustrated in Figure 2. The final dataset consists of 354 fall files and 899 non-fall files. The dataset is publicly available in the following repository[1].

3.3 Hyperparameters Tuning

The hyperparameter values are used to control the learning process of the ML model. Despite using the same training data, varying hyperparameter values lead to distinct trained models. The process of selecting the most effective combination of hyperparameter values is referred to as hyperparameter tuning and holds significant importance in attaining high model performance [30].

Unfortunately, there is not a one-size-fits-all set of optimal hyperparameters for all problems, and evaluating different combinations of hyperparameter values is computationally expensive. Nevertheless, hyperparameter tuning strategies and commonly employed hyperparameter values exist that have proven successful in addressing similar problems. In our experiments, we employed the widely used grid-search strategy for hyperparameter tuning in Support Vector Machines and Random Forests.

This strategy involves systematically selecting various hyperparameter values and evaluating all possible configurations. The grid-search process was implemented using the GridSearchCV function provided by the Scikit-learn Python library. This approach allows for a comprehensive exploration of hyperparameter combinations to identify the most effective configuration for our specific experiments [22].

Due to the consideration of a larger number of hyperparameters in the CNN architecture, we employed the hyperband method for hyperparameter tuning [15]. The hyperband method extends the successive halving algorithm [10], and its process is outlined as follows: a set of n hyperparameter values is evaluated for all configurations using limited resources (e.g., dataset size, training time, number of epochs).

After evaluation, the configurations with the worst performance are discarded, and the process is iterated until only the best configuration remains. Unlike the successive halving algorithm, the hyperband method allocates a specific number of iterations for different configurations, focusing on promising candidates for more extensive evaluations. In our work, hyperband tuning was implemented using the Keras hyperparameter tuning library [21]. Table 1 outlines the values considered for hyperparameter tuning in each machine learning model.

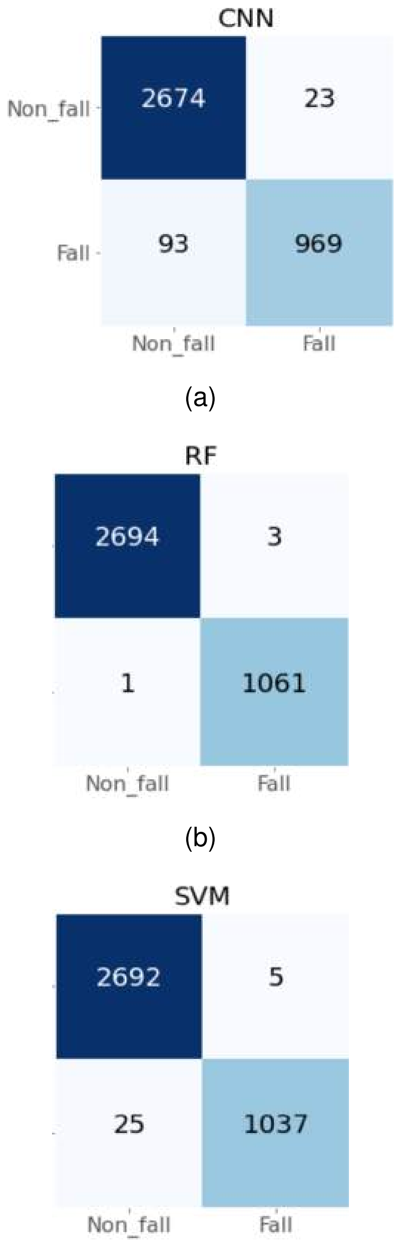
4 Results and Discussions

Following hyperparameter tuning, the best configurations for the three machine learning models are described as follows. For the SVM classifier, the best hyperparameter configuration was $\{C = 10$, kernel=linear$\}$.

For the RF model, the best hyperparameter configuration was {non-bootstrap, max_depth=50, min_samples_leaf = 3, n_estimators = 100}.

Lastly, for the CNN architecture, we obtained a three-convolutional layer network followed by two fully connected layers. The first convolutional layer consists of 32 filters (3×3) with a ReLU activation function followed by a max pooling layer (2×2). The second and third convolutional layers consist of 64 and 192 filters (3×3), respectively, followed by an average pooling layer (2×2). After convolutional layers, a fully connected layer of 32 nodes and a tanh activation function were added.

[1] github.com/rosendo655/fall_data_analysis



(a)

(b)

(c)

Fig. 4. Confusion matrices of the evaluated models: (a) CNN, (b) RF, and (c) SVM

The output layer has two nodes and a softmax activation function.

Figure 3 shows the CNN architecture[2]. These configurations represent the best-performing settings after thorough hyperparameter tuning for each respective machine-learning model.

The training of the CNN architecture involved the utilization of the cross-entropy loss function and the Adam optimization algorithm [13]. The model was trained with a learning rate set to 10^{-3} and a batch size of 16 throughout 30 epochs. These parameters were chosen to optimize the training process and achieve effective learning for the given dataset.

The performance results obtained by the models were assessed through repeated 3×10-fold cross-validation [11]. The evaluation employed standard performance metrics for machine learning models [11], including Accuracy (ACC), Balanced Accuracy (BACC), and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics offer a comprehensive evaluation of the overall method performance.

Table 2 shows the average results of the evaluated ML techniques alongside related literature works. It is evident that the CNN, SVM, and RF models exhibit robust capabilities in distinguishing between falls and non-fall events, achieving performances consistently above 95% across all metrics.

In comparison to the works of Mashiyama et al. [17] and Taniguchi et al. [26] works, our evaluated machine learning models demonstrated superior performances, exceeding 96% accuracy with only one sensor and a lower sensor resolution.

Notably, the RF classifier achieved outstanding performance, reaching up to 99% for ACC, BACC, and AUC-ROC. Moreover, Figure 4 illustrates that the RF model incurred only four errors, primarily misclassifying non-fall data as fall data.

Importantly, misclassifying a non-fall event as a fall event is often considered less critical than the opposite scenario. This emphasizes the effectiveness of the RF model in minimizing errors and underscores its potential as a reliable fall detection solution.

[2] This figure was generated by adapting the code from: github.com/gwding/draw_convnet

5 Conclusion and Future Work

This work addresses the problem of elderly fall event detection using a thermal sensor and machine learning techniques. Unlike other devices, the thermal sensor is less invasive, has no need to be manipulated, and maintains privacy. Additionally, the sensor's low resolution opens up possibilities for embedded applications, enhancing its versatility in various contexts.

The experimental dataset utilized in this study was generated by students through simulations of both fall and non-fall events, during which a thermal sensor recorded room temperatures. The captured data for each event was subsequently stored in a CSV file. The compiled dataset comprises a total of 1,253 CSV files, consisting of 354 fall events and 899 non-fall events. Each file is structured with 25 rows (frames) and 16 columns, representing the recorded room temperatures during the respective events.

For the fall event detection, we selected three well-known supervised machine learning models that have shown outstanding performance for several machine learning tasks: SVM, RF, and CNN. The experimental outcomes affirm the efficacy of the CNN, SVM, and RF models in effectively distinguishing between fall and non-fall events. Notably, the random forest classifier demonstrated the most favorable results in 3×10-Fold cross-validation, with merely four errors. This underscores the remarkable detection capabilities achievable with just one sensor.

The findings highlight the potential of leveraging these machine-learning models for reliable and efficient fall detection using minimal sensor resources. In future research works, it would be beneficial to incorporate more complex scenes during the training stage, encompassing scenarios involving multiple individuals or pets.

This approach aims to enhance the robustness of the fall detection system by exposing it to a broader range of environmental conditions. By training the system on diverse and challenging scenarios, it can develop a more comprehensive understanding of potential fall events in real-world settings, thus improving its reliability and applicability across various contexts.

Acknowledgments

In memory of Dr. Arnoldo Díaz Ramírez, an outstanding professor, cherished friend, and, above all, a remarkable human being. This work is funded by CONAHCYT.

References

1. **Bergen, G., Stevens, M. R., Burns, E. R. (2016).** Falls and fall injuries among adults aged $\geq$ 65 years — United States, 2014. Morbidity and Mortality Weekly Report, Vol. 65, No. 37, pp. 993–998. DOI: 10.15585 /mmwr.mm6537a2.

2. **Breiman, L. (2001).** Random forests. Machine Learning, Vol. 45, No. 1, pp. 5–32. DOI: 10.1 023/A:1010933404324.

3. **Chandra, M. A., Bedi, S. S. (2018).** Survey on SVM and their application in image classification. International Journal of Information Technology, Vol. 13, No. 5, pp. 1–11. DOI: 10.1007/s41870-017-0080-1.

4. **Cortes, C., Vapnik, V. (1995).** Support-vector networks. Machine Learning, Vol. 20, No. 3, pp. 273–297. DOI: 10.1007/BF00994018.

5. **Diaz-Ramirez, A., Dominguez, E., Martinez-Alvarado, L. (2015).** A falls detection system for the elderly based on a WSN. Proceedings of the IEEE International Symposium on Technology and Society, IEEE, pp. 1–6. DOI: 10.1109/istas.2015.7439426.

6. **Haddad, Y. K., Bergen, G., Florence, C. S. (2019).** Estimating the economic burden related to older adult falls by state. Journal of Public Health Management and Practice, Vol. 25, No. 2, pp. E17–E24. DOI: 10.1097/ phh.0000000000000816.

7. **Heaton, J. (2017).** Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning, Vol. 19. Springer Science and Business Media LLC. DOI: 10.1007/s10710-017-9314-z.

8. **Ho, T. K. (1995).** Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition, pp. 278–282. DOI: 10.1109/ICDAR.1995.598 994.

9. **Hung, D. H., Saito, H., Hsu, G. S. (2013).** Detecting fall incidents of the elderly based on human-ground contact areas. Proceedings of the 2nd IAPR Asian Conference on Pattern Recognition, pp. 516–521. DOI: 10.1109/AC PR.2013.124.

10. **Jamieson, K., Talwalkar, A. (2016).** Non-stochastic best arm identification and hyperparameter optimization. Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Vol. 41, pp. 240–248. DOI: 10.48550/arXiv.1502.07943.

11. **Japkowicz, N., Shah, M. (2011).** Evaluating learning algorithms: A classification perspective. Cambridge University Press. DOI: 10.1017/CBO9780511921803.

12. **Jia, H., Lubetkin, E. I., DeMichele, K., Stark, D. S., Zack, M. M., Thompson, W. W. (2019).** Prevalence, risk factors, and burden of disease for falls and balance or walking problems among older adults in the U.S. Preventive Medicine, Vol. 126, pp. 105737. DOI: 10.101 6/j.ypmed.2019.05.025.

13. **Kingma, D. P., Ba, J. (2014).** Adam: A method for stochastic optimization. arXiv. DOI: 10.485 50/ARXIV.1412.6980.

14. **Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998).** Gradient-based learning applied to document recognition. Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324. DOI: 10 .1109/5.726791.

15. **Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A. (2017).** Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research, Vol. 18, No. 1, pp. 6765–6816.

16. **Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2021).** A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, No. 12. DOI: 10.1109/TNNLS.2021.3084827.

17. **Mashiyama, S., Hong, J., Ohtsuki, T. (2014).** A fall detection system using low resolution infrared array sensor. Proceedings of the IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication, pp. 2109–2113. DOI: 10.110 9/pimrc.2014.7136520.

18. **Mecocci, A., Micheli, F., Zoppetti, C., Baghini, A. (2016).** Automatic falls detection in hospital-room context. Proceedings of the 7th IEEE International Conference on Cognitive Infocommunications, pp. 127–132. DOI: 10.1 109/coginfocom.2016.7804537.

19. **Mubashir, M., Shao, L., Seed, L. (2013).** A survey on fall detection: Principles and approaches. Neurocomputing, Vol. 100, pp. 144–152. DOI: 10.1016/j.neucom.2011. 09.037.

20. **Nishio, K., Kaburagi, T., Hamada, Y., Matsumoto, T., Kumagai, S., Kurihara, Y. (2021).** Construction of an aggregated fall detection model utilizing a microwave doppler sensor. IEEE Internet of Things Journal, Vol. 9, No. 3, pp. 2044–2055. DOI: 10.1109/JIOT.202 1.3089520.

21. **O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L. (2019).** Kerastuner. github.com/keras-team/keras-tuner.

22. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011).** Scikit-learn: Machine learning in python. Journal of Machine Learning Research, Vol. 12, No. 85, pp. 2825–2830.

23. **Ren, L., Peng, Y. (2019).** Research of fall detection and fall prevention technologies: A systematic review. IEEE Access, Vol. 7, pp. 77702–77722. DOI: 10.1109/ACCESS.2 019.2922708.

24. **Rubenstein, L. Z. (2006).** Falls in older people: Epidemiology, risk factors and strategies for prevention. Age and Ageing, Vol. 35, No. suppl 2, pp. ii37–ii41. DOI: 10.1093/ageing/afl084.

25. **Shaik, A. B., Srinivasan, S. (2018).** A brief survey on random forest ensembles in classification model. Proceedings of the International Conference on Innovative Computing and Communications, pp. 253–260. DOI: 10.1007/978-981-13-2354-6_27.

26. **Taniguchi, Y., Nakajima, H., Tsuchiya, N., Tanaka, J., Aita, F., Hata, Y. (2014).** A falling detection system with plural thermal array sensors. Proceedings of the Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced Intelligent Systems, pp. 673–678. DOI: 10.1109/SCIS-ISIS.2014.7044834.

27. **Wang, X., Ellul, J., Azzopardi, G. (2020).** Elderly fall detection systems: A literature survey. Frontiers in Robotics and AI, Vol. 7, pp. 71. DOI: 10.3389/frobt.2020.00071.

28. **World Health Organization (2024).** Ageing and health. www.who.int/news-room/fact-sheets/detail/ageing-and-health.

29. **World Health Organization (2024).** Falls. www.who.int/news-room/fact-sheets/detail/falls.

30. **Yang, L., Shami, A. (2020).** On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, Vol. 415, pp. 295–316. DOI: 10.1016/j.neucom.2020.07.061.

31. **Zhang, Z., Conly, C., Athitsos, V. (2015).** A survey on vision-based fall detection. Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, pp. 1–7. DOI: 10.1145/2769493.2769540.

Article received on 28/02/2024; accepted on 10/09/2024.
**Corresponding author is Julia Diaz-Escobar.*