engineering proceedings

Proceeding Paper

Deep Transfer Learning Approach in Smartwatch-Based Fall Detection Systems †

Alessandro Leone *, Andrea Manni, Gabriele Rescio, Pietro Siciliano and Andrea Caroppo

National Research Council of Italy, Institute for Microelectronics and Microsystems, 73100 Lecce, Italy; andrea.manni@cnr.it (A.M.); gabriele.rescio@cnr.it (G.R.); pietroaleardo.siciliano@cnr.it (P.S.); andrea.caroppo@cnr.it (A.C.)

* Correspondence: alessandro.leone@cnr.it

† Presented at the 1st International Conference on AI Sensors & the 10th International Symposium on Sensor Science, Singapore, 1–4 August 2024.

Abstract: This study introduces a fall detection system utilizing an affordable consumer smartwatch and smartphone with edge computing capabilities for implementing AI algorithms. Due to the widespread use of these devices, the system as a whole is extremely accepted, easy to use, requires no tuning of any kind, and guarantees extended functioning for a long period. From a technical standpoint, falls are identified using AI techniques to analyze 3D raw data acquired by the smartwatch's built-in accelerometer. However, existing AI models for fall detection are often trained on simulated falls involving young people, which may not accurately represent the falls of elderly in unhealthy conditions, such as arthritis or Parkinson's disease, leading to limitations in detecting falls in this population. Additionally, variations in hardware features among different smartwatches can result in inconsistencies in accelerometer data measurements across X, Y, and Z orientations, further complicating accurate fall detection. To address the challenge of limited and device-specific datasets and to enhance model generalization across various devices, a Deep Transfer Learning approach is proposed. This method proves effective when data are poor. Specifically, the Continuous Wavelet Transform (CWT) is applied to raw accelerometer signals to convert them into 2D images, enabling the use of deep architectures for Transfer Learning. By employing CWT on 5 s time windowed raw accelerometer signals, heat maps (scalograms) are generated. Real-time accelerations sampled at 50 Hz are collected using a smartwatch application, transmitted via Bluetooth to a smartphone app, and converted into scalograms. These serve as input for pre-trained Deep Learning models to estimate fall probabilities. Preliminary tests on the Wrist Early Daily Activity and Fall Dataset (WEDA-FALL) show promising results with an accuracy of approximately 98%, underscoring the efficacy of utilizing wrist-worn wearable devices for processing raw accelerometer data.

Keywords: transfer learning; deep learning; fall detection; smartwatch

check for updates

Citation: Leone, A.; Manni, A.; Rescio, G.; Siciliano, P.; Caroppo, A. Deep Transfer Learning Approach in Smartwatch-Based Fall Detection Systems. *Eng. Proc.* **2024**, *78*, 2. https://doi.org/10.3390/engproc2024078002

Academic Editor: Po-Liang Liu

Published: 18 November 2024

1. Introduction

According to the latest report of the World Health Organization (WHO) [1], an average of 37.5 million falls require medical assistance each year. In particular, about 30% of the population over-65s falls each year, reaching 40% in the over-70s. Falling is the second most frequent reason for accidental death worldwide, and adults over 60 years of age are the most affected [1,2]. So, for this kind of user, a fall is a critical and unsafe situation requiring rapid assistance to reduce the consequences from health, social, welfare, and economic points of view. It has been shown [3,4] that elderly people having previous falls develop a deep-seated anxiety of falling, resulting in a decrease in their level of mobility and quality of life itself.

In the last 15 years, fall detection systems received significant focus from the scientific community, and several systems [5–8] have been implemented (often integrating simple threshold-based or rules-based approaches [9,10] with different detection rates and levels of acceptability/usability). Presently, two groups of fall detection sensors have been identified in the scientific community. A first group of fall detection sensors is based on ambient devices, such as floor pressure sensors [11,12], acoustic sensors [13], video cameras [14–16], etc. Such devices are expensive; sensitive to surrounding conditions such as, for example, uniform lighting and the presence of furniture in the environment, which can lead to false alarms and decrease their accuracy [17]; and are dependent on number and positioning [7]. Furthermore, they may be subject to privacy issues inhibiting their use in the home (especially in bedrooms and bathrooms, where many falls typically occur). The second group involves wearable sensor-based methods used to evaluate speed and unexpected position changes. Usually, accelerometers, gyroscopes, or magnetometers are employed for fall detection [18–20], also using commercially available devices [21–23]. In contrast to ambient sensors, wearable sensors have clear benefits such as user-friendliness, size, and low cost, due to the presence of these sensors in devices already available to end-users (e.g., smartphones and wristbands). Several commercial smartwatches have technology to support elderly with health problems, detecting the main activities of daily living (ADLs) such as walking, sitting and lying down.

2. Related Works

Fall detection via smartwatches is becoming increasingly popular in recent years, as evidenced by the number of recent scientific publications using different algorithimic approaches. In [24], a fall detection solution is proposed, consisting of a smartwatch paired with a smartphone, a cloud-based storage system, and data analysis packages using a Machine Learning (ML) approach training a Support Vector Machine (SVM) algorithm using the following features: the resulting minimum and maximum acceleration in a sliding window of 750 ms; the Euclidean norm of the difference between the maximum and minimum acceleration in the same sliding window; and the acceleration vector length at the time of sampling. An average accuracy of 93.8% was obtained with this model. The main limitation of this study is that only data from young and healthy individuals were used for training. In [25], a Deep Learning (DL) strategy to effectively detect four different types of fall and nine common ADLs was investigated. The performance of four different DL models and six ML models in classifying different human activities other than falls is compared, and Gated Recurrent Units (GRU) model was selected as the best model, with an average accuracy of around 90.5% on a dataset proposed in the same paper which, however, contains data on falls performed by seven young subjects aged between 21 and 55, which presents generalization limitations. A DL approach for fall detection is also introduced in [26]. First, a data augmentation methodology is used due to data imbalance. Subsequently, a BiLSTM network is trained using, as input, (1) only the acceleration values, (2) only the gyroscope values, and (3) their combination. The paper reports both the obtained accuracy in classifying specific ADLs (sitting, squatting, walking, running) and the obtained accuracy to differentiate the fall from the activities. The achieved results show that the fall can only be classified using the measured acceleration from the smartwatch. However, also in this work, data were acquired with end-users aged between 19 and 25 years. A very interesting approach is presented in [27] using a combination of smartwatch and smartphone. In particular, an Android application called "SmartFall" is implemented that acquires accelerometer data from a smartwatch, and processes it using a DL architecture. Two standard ML algorithms, SVM and Naive Bayes (NB), are compared with a GRU model. Training was performed using data from three different benchmark datasets containing more general training data compared to previously work in the literature. The highest average accuracy obtained was 79%. Additionally, the related studies were summarized according to the used methodology, fall classification accuracy, and limitations, as shown in Table 1.

Table 1. Summary of related studies.

Paper	Method	Accuracy (%)	Limitations
[24]	SVM	93.8	Only data from young and healthy individuals
[25]	GRU	90.5	Data from seven young subjects
[26]	BiLSTM	97.35	End-users aged between 19 and 25 years
[27]	GRU	79	Not high accuracy, but more general training data

As highlighted, most of the proposed work has limitations in data type used to train learning patterns, since young subjects are normally used to replicate ADLs and falls. However, it is known that the movements of the elderly can be affected due to motor pathologies that increase violent movements in the smartwatch, which can lead to the generation of patterns in the accelerometer data uncorrelated with falling. To overcome the problem of the lack of datasets related to falls by elderly subjects, the proposed work introduces the application of a Transfer Learning (TL) strategy applied to one-dimensional signals (e.g., raw accelerometers on three axes) and based on the use of the Continuous Wavelet Transform (CWT). Using the CWT, the classification problem of 1D signals is reduced to the image classification (the typical input of a deep architecture used for TL), representing the same input signal in two dimensions. For this purpose, in order to more accurately classify the fall event, three different pre-trained architectures among the most widely used in the literature are compared: DenseNet201, VGG16, and ResNet50.

The remainder of this paper is organized as follows. Section 3 reports some details on the implemented algorithmic pipeline. The preliminary results are included in Section 4, while the conclusions are reported in Section 5.

3. Materials and Methods

In this paper, an algorithmic pipeline for real-time fall detection using the raw accelerometer data returned by any consumer smartwatch is introduced. The architecture of the proposed system is shown in Figure 1.

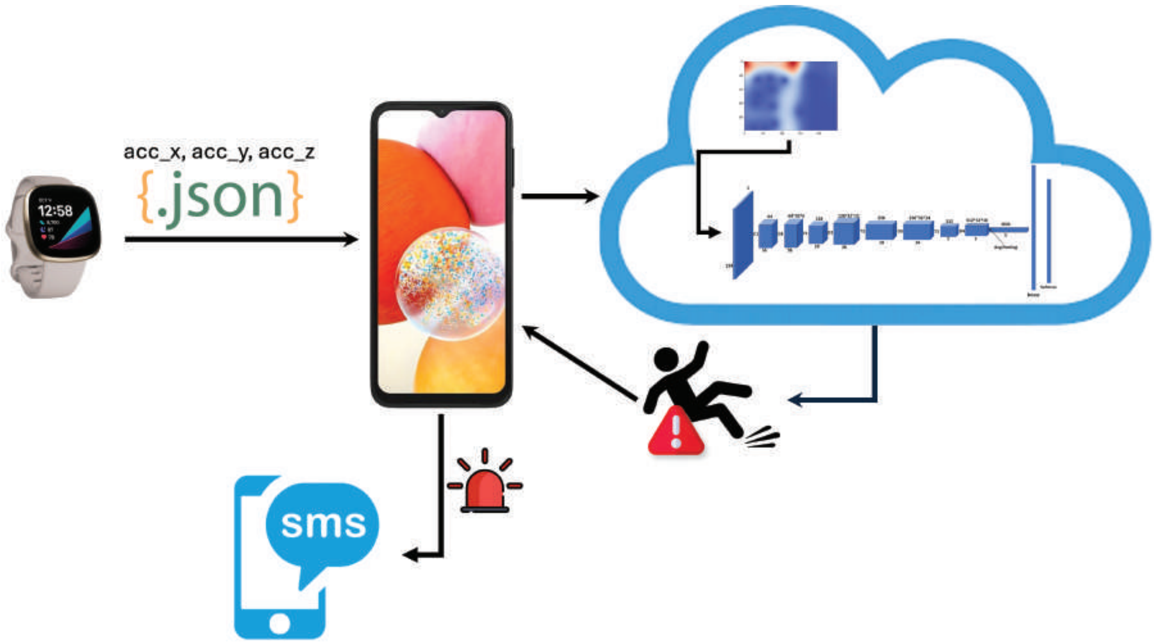

Figure 1. Overview of the proposed fall detection system using raw accelerometer data obtained from a commercial smartwatch.

First, an Android application was developed and installable on any commercial smartphone to collect estimated acceleration values from the smartwatch in real time, along all three axes. The app acts as a server, receiving data from an app installed on the smartwatch reading the raw accelerometer data sampled at 50 Hz and encapsulating it in a JSON message in 5 s packets. The collected data from the smartwatch are sent to the cloud, for use by any processing unit on which TL's fall detection strategy is implemented. More details can be found in the following subsections. In IoT applications, data privacy is a major issue in healthcare monitoring systems. In the developed application, the collected data are all de-identified and indexed with a user id, whose association to the end-user is only known to the associated caregiver.

As described above, the goal is to use this approach on any commercial smartwatch that can be paired with Android-based smartphones. However, at this prototyping stage, the Fitbit Sense [28] was chosen as the wristwatch device due to the variety of supported sensors and its low cost. In fact, this smartwatch is capable of detecting heart rate, electrodermal activity, skin temperature, sleep monitoring, respiratory rate, SpO2, GPS position and, of course, it has an accelerometer and a gyroscope, from which it is possible to acquire a different sampling rate. The Samsung Galaxy A32 5G smartphone [29] was selected to execute our fall detection app, and to receive sensor data from the smartwatch via a Bluetooth communication protocol. It has a 2GHz octa-core processor and 4 gigabytes of RAM that allows for real-time data reception and processing over a long period.

3.1. Accelerometer Signals Processing Technique

As described in Section 1, DL strategies have been increasingly used for fall detection algorithms in recent years, although classical approaches, such as SVM and Decision Trees (DT), are still more popular due to the limited size of fall detection datasets. However, artificial neural networks avoid the feature extraction and selection phase, which can be a laborious task. Furthermore, the problem of small data size can be solved using TL techniques [30], involving the use of pre-trained Convolutional Neural Networks (CNN).

Since such networks require images as input, it is necessary to transform 1-D accelerometer signals into 2-D images. To achieve this, various approaches exist. In this work, CWT was applied, which allows us to obtain a time-frequency representation (a scalogram) of the signal, and to simultaneously capture its peculiarities in different frequency bandwidths, which is useful to process non-stationary signals.

For a time series $x(t)$, CWT is expressed as follows:

$$X_{\omega}(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\varphi^{*}\left(\frac{t-b}{a}\right)dt \tag{1}$$

where a and b are the scaling and translational value, respectively ($a > 0$, a and $b \in \mathbb{R}$), while $\varphi^{*}(t)$ is the complex conjugate of the analyzed mother wavelet and, finally, $1/\sqrt{a}$ is a normalization factor [31]. Applying CWT to time windows of accelerometer signals, a heat map known in the scientific literature as a scalogram can be obtained. A scalogram represents the absolute value of the CWT coefficients of a signal, with the b and a values along the x- and y-axis, respectively, and the intensity of each point measured by Equation (1). In the proposed approach, the scalograms were extracted from the time series of the three raw accelerometer signals split into 5 s time windows. Normally, a smaller value of the scaling quantity enables better evaluation of abrupt changes, while a higher value provides more information that can lead to better classification accuracy. The most commonly used scaling values in this context are 32, 64, 128, and 256. In the present work, the scaling value was set to 256, while Morlet was used as a wavelet. The resulting scalograms are then resized to 224 $\times$ 224 to fit with the input layer of the selected deep architectures shown in the next section.

Figure 2 shows an example of a scalogram obtained for walking (a) and fall (b) with the corresponding raw accelerometer signals, respectively. For the sake of brevity, only x-axis of the accelerometer signal is reported.

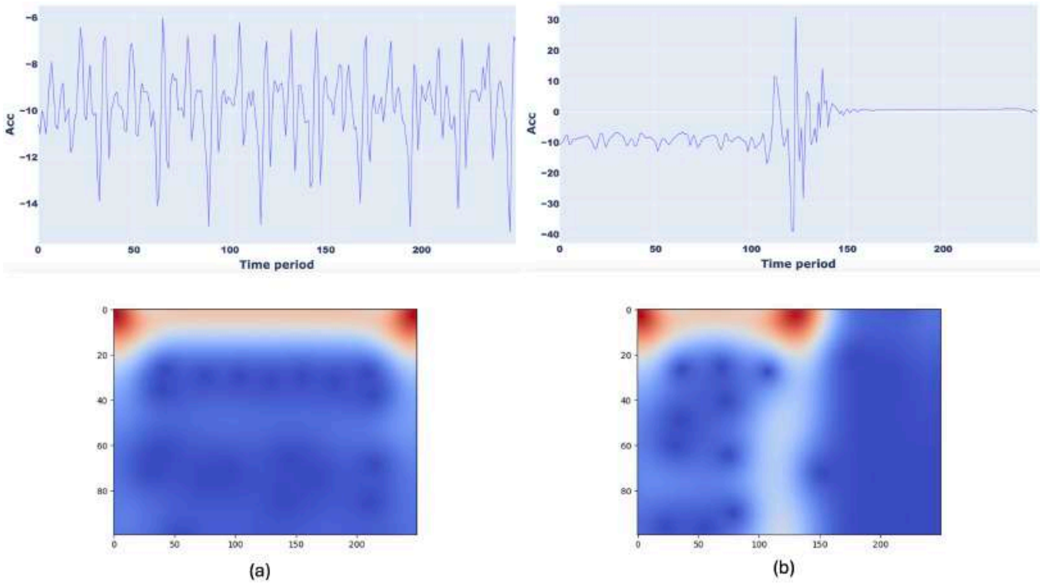

Figure 2. Generated scalogram images (scale factor = 256 and wavelet = Morlet) from the raw accelerometer *x*-axis signal for (**a**) walking activity and (**b**) fall event.

3.2. Model Selection for Transfer Learning

CNN architectures suffer from a major drawback, namely the need for a large amount of data for training purposes. This problem is more evident when decisions must be performed on rare data, as for fall detection. To overcome this problem, a TL strategy is applied in our proposed fall detection system. The main idea of TL is to extract knowledge from a dataset (i.e., "source domain") and then transfer it to a new dataset (i.e., "target domain") to improve learning [32]. In the fall detection context via accelerometer data, the principal reason to use TL is the lack of much data in the application domain, especially falls of elderly subjects. Furthermore, the annotation of these data are expensive and sometimes subjective, especially for areas requiring experts for data labeling and with a high inter-/intra-class variability.

To select a CNN model, three popular architectures were tested, namely DenseNet201, VGG16, and ResNet50 which, in the literature, have achieved important results in the classification of scalogram images. All of these models were trained on all generated scalograms. Each input sample consisted of 3 scalograms (one scalogram for each signal channel).

Each network was modified and tuned for the fall detection problem. Each of the original networks was trained to classify images for 1000 classes on the ImageNet dataset, but in this work, there is a binary classification problem (Fall/No Fall). Therefore, the last fully connected layer was replaced with a new classification layer structured through six new blocks (flatten:1 and dense:5) producing two outputs (fall and no fall). In Table 2, optimized hyperparameters for the three trained models are shown.

Table 2. Hyperparameters for the proposed TL models.

Hyperparameter	**Model Architecture**		
	DenseNet201	**VGG16**	**ResNet50**
Learning rate	0.002	0.001	0.002
Batch size	128	128	128
Optimizer	Adam	Adam	Adam
Output activation layer	softmax	softmax	softmax
Number of epochs	50	70	50

For the sake of brevity, Figure 3 shows the graphical representation of the proposed TL scheme for the DenseNet201 network only.

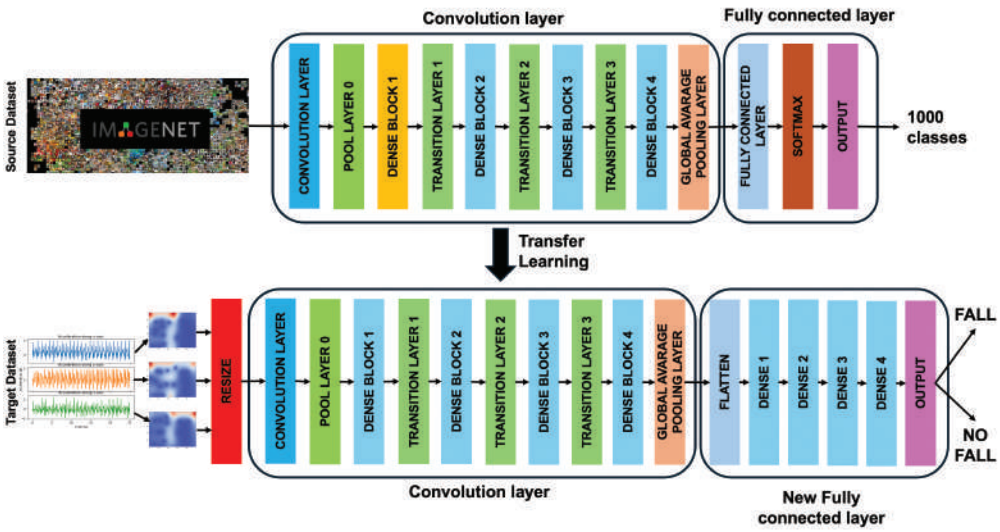

Figure 3. Proposed DenseNet201 architecture for fall detection.

3.3. Dataset

To validate the proposed architecture, a series of tests were performed using the Wrist Early Daily Activity and Fall Dataset (WEDA) [33] as benchmark dataset. It uses the commercial Fitbit Sense® smartwatch to provide accelerometric and gyroscopic data on 25 users (14 young and 11 elderly over 80). Data are sampled at different frequencies from 5 to 50 Hz and labeled with 8 types of fall and 11 ADLs, shown in Table 3.

Table 3. Types of falls and ADLs in the considered dataset.

Types of Falls	Types of ADL
Fall forward while walking caused by a slip	Walking
Lateral fall while walking caused by a slip	Jogging
Fall backward while walking caused by a slip	Walking up and downstairs
Fall forward while walking caused by a trip	Sitting on a chair, wait a moment, and get up
Fall backward when trying to sit down	Sitting a moment, attempt to get up and collapse into a chair
Fall forward while sitting, caused by fainting or falling asleep	Crouching (bending at the knees), tie shoes, and get up
Fall backward while sitting, caused by fainting or falling asleep	Stumble while walking
Lateral fall while sitting, caused by fainting or falling asleep	Gently jump without falling (trying to reach high object)
	Hit table with hand
	Clapping Hands
	Opening and closing door

In our tests, accelerometer data sampled at 50 Hz were used, and training experiments were performed on a Dell™ Precision 7920 Rack workstation, with 256GB RAM, dual Intel Xeon Gold 5218R CPU@2.10Ghz processors, three NVIDIA™ RTX A2000 12 GB GPUs, using Python version 3.8 and the following libraries: Tensorflow (2.10), pandas (2.0.3), scikit-learn (1.2.1), spkit (0.0.9.6.7).

4. Results and Discussions

Firstly, to achieve the objectives of this study, and to evaluate the goodness of the proposed approach in fall detection, three different models (DenseNet201, VGG16, and ResNet50) were trained. The performance of these models were evaluated using a 10-cross validation [34]. Consequently, each model was trained with 80% of the data, and testing was performed with the remaining 20% and, to avoid over-fitting, 10% of the testing set was used as validation. The entire process was replicated 10 times, using different training and testing sets to prevent the occurrence of the same samples in the training and testing set at the same time. Four well-known metrics were calculated for the numerical evaluation of performance, such as accuracy, precision, recall, and F1 score. Table 4 shows the obtained results for each considered model, demonstrating the goodness of the proposed approach, with an average accuracy in the range of 95.24% for VGG16 to 97.56% for DenseNet201.

Table 4. Comparison of the performance for each model.

Model	Accuracy	Precision	Recall	F1-Score
DenseNet201	0.9756	0.9743	0.9674	0.9741
VGG16	0.9524	0.9614	0.9520	0.9686
ResNet50	0.9619	0.9713	0.9620	0.9750

For brevity, Figure 4 shows, respectively, the loss and accuracy during training and validation of our best performing model. In particular, from Figure 4, it is evident how our model improves at varying epochs, fixed at 50 in this work, while Figure 5 reports the confusion matrices of the obtained average accuracies for each considered model.

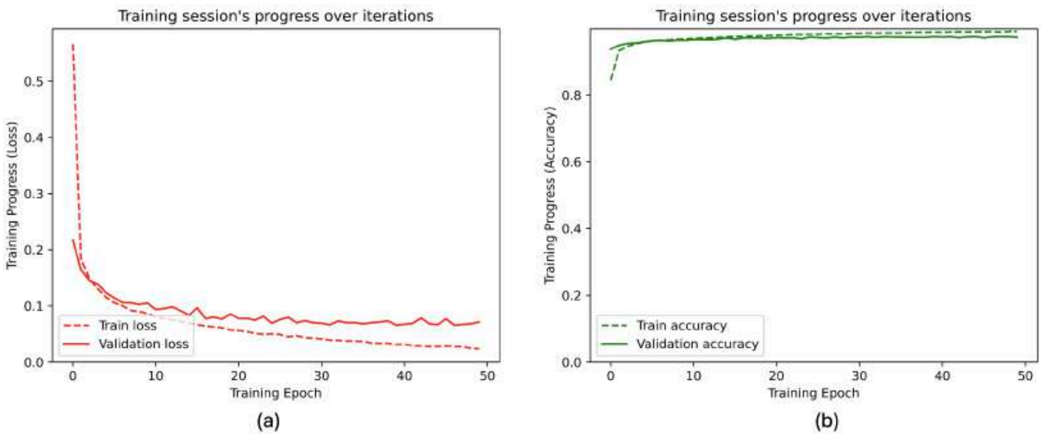

Figure 4. (**a**) Loss and (**b**) accuracy of the proposed DenseNet201 at training and validation phase.

Furthermore, Table 5 compares the classification performance (in terms of accuracy) obtained with the proposed approach with two other research papers based on the same dataset. As can be seen, the obtained accuracy is similar to [35], with the advantage of a high degree of generalization approach due to the TL usage and, consequently, with a greater adaptability in real-world contexts.

Table 5. Comparison of the performance for each model.

Model	Accuracy
LightGBM [36]	0.9530
kNN [35]	0.9805
Our DenseNet201	0.9756

Furthermore, in [35] kNN is used, a supervised learning method requiring a large amount of labeled training data with the difficulty of classifying against new data, resulting in overfitting. Finally, another obvious difference between [35] and our approach is the duration of the sliding window (9 s versus 5 s), a very important parameter given the need to detect a fall in the shortest possible time.

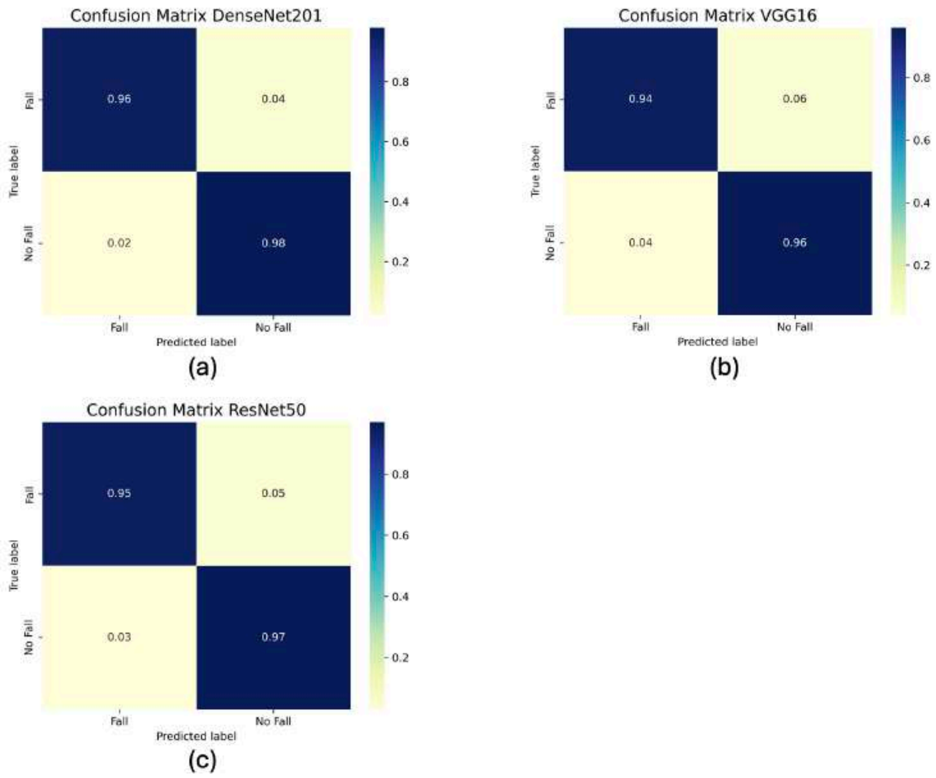

Figure 5. Confusion matrix for the proposed models: (**a**) DenseNet201, (**b**) VGG16, (**c**) ResNet50.

5. Conclusions

In this work, a system for automatic fall detection was designed and implemented, using a commercial smartwatch for data collection and a smartphone as the processing unit. TL was used for the analysis of accelerometric data extracted from the wearable device, due to the lack of large amounts of data in the considered application context. Three different pre-trained models were employed and it was shown that the DenseNet201 model performed best in terms of accuracy in classifying the different types of analyzed falls.

However, the proposed system has limitations compared to commercial solutions already on the market. The most important limitation derives from the use of the cloud, increasing its cost and making it impossible to detect a fall if there is no network connection. A second limitation relates to the impossibility of sending a fall detected in the event of no network connection to the smartphone. A further limitation is the necessary proximity of the smartphone to the smartwatch due to the Bluetooth connection required for the correct transmission of the raw data.

Possible future developments will focus on the use of additional benchmark datasets (as MobiFall Dataset [37], SisFall [38], and tFall [39]) to validate the algorithmic methodology's performance, try to reduce the cost of the entire solution, and for methodology implementation directly on the smartphone using the network model previously trained on high-performance computing hardware to avoid the use of the cloud, minimizing possible data transmission and privacy issues. In addition, further deep architectures will be compared in addition to the three considered to evaluate whether DenseNet201 continues to be the best choice in terms of overall accuracy. In the end, to improve the quality of the results and to estimate the performance on new subjects, a leave-one-out subject cross-validation will be implemented.

Author Contributions: Conceptualization, A.L, A.M., G.R. and A.C.; methodology, A.L., A.M., G.R. and A.L.; validation and experimental investigation, A.L., A.M., G.R. and A.C.; writing—original draft preparation, A.L., A.M., G.R. and A.C.; review and editing, A.L., A.M., G.R. and A.C.; supervision, A.L. and P.S. All authors have read and agreed to the published version of the manuscript

Funding: This paper was developed within the project funded by Next Generation EU-"Age-It-Ageing well in an ageing society" project (PE0000015), National Recovery and Resilience Plan (NRRP)-PE8-Mission 4, C2, Intervention 1.3. The views and opinions expressed are only those of the authors and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available upon request due to privacy restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. WHO. Falls, Fact Sheet. 2021. Available online: https://www.who.int/en/news-room/fact-sheets/detail/falls (accessed on 3 July 2024).
2. World Health Organization; Ageing and Life Course Unit. *WHO Global Report on Falls Prevention in Older Age*; World Health Organization: Geneva, Switzerland, 2008.
3. Legters, K. Fear of falling. *Phys. Ther.* **2002**, *82*, 264–272. [CrossRef] [PubMed]
4. Vellas, B.J.; Wayne, S.J.; Romero, L.J.; Baumgartner, R.N.; Garry, P.J. Fear of falling and restriction of mobility in elderly fallers. *Age Ageing* **1997**, *26*, 189–193. [CrossRef] [PubMed]
5. Wang, X.; Ellul, J.; Azzopardi, G. Elderly fall detection systems: A literature survey. *Front. Robot.* **2020**, *7*, 71. [CrossRef]
6. Thakur, N.; Han, C.Y. A study of fall detection in assisted living: Identifying and improving the optimal machine learning method. *J. Sens. Actuator Netw.* **2021**, *10*, 39. [CrossRef]
7. Singh, A.; Rehman, S.U.; Yongchareon, S.; Chong, P.H.J. Sensor technologies for fall detection systems: A review. *IEEE Sens. J.* **2020**, *20*, 6889–6919. [CrossRef]
8. Yhdego, H.; Paolini, C.; Audette, M. Toward Real-Time, Robust Wearable Sensor Fall Detection Using Deep Learning Methods: A Feasibility Study. *Appl. Sci.* **2023**, *13*, 4988. [CrossRef]
9. Miaou, S.G.; Shih, F.C.; Huang, C.Y. A smart vision-based human fall detection system for telehealth applications. In Proceedings of the Third IASTED International Conference on Telehealth (Telehealth07), Montreal, QC, Canada, 30 May–1 June 2007; pp. 7–12.
10. Sase, P.S.; Bhandari, S.H. Human fall detection using depth videos. In Proceedings of the 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, Delhi, 22–23 February 2018; pp. 546–549.
11. Clemente, M.; Li, F.; Valero, M.; Song, W. Smart Seismic Sensing for Indoor Fall Detection, Location, and Notification. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 524–532. [CrossRef]
12. Huang, Y.; Chen, W.; Chen, H.; Wang, L.; Wu, K. G-Fall: Device-free and Training-free Fall Detection with Geophones. In Proceedings of the 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 10–13 June 2019; pp. 1–9.
13. Vacher, M.; Bouakaz, S.; Bobillier-Chaumon, M.-E.; Aman, F.; Khan, R.A.; Bekkadja, S.; Portet, F.; Guillou, E.; Rossato, S.; Lecouteux, B. The CIRDO Corpus: Comprehensive Audio/Video Database of Domestic Falls of Elderly People. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia, 23–28 May 2016; pp. 1389–1396.
14. Fatima, M.; Yousaf, M.H.; Yasin, A.; Velastin, S.A. Unsupervised fall detection approach using human skeletons. In Proceedings of the 2021 International Conference on Robotics and Automation in Industry (ICRAI), Rawalpindi, Pakistan, 26–27 October 2021; pp. 1–6.
15. Ramirez, H.; Velastin, S.A.; Meza, I.; Fabregas, E.; Makris, D.; Farias, G. Fall Detection and Activity Recognition Using Human Skeleton Features. *IEEE Access* **2021**, *9*, 33532–33542. [CrossRef]
16. Asif, U.; Mashford, B.; Cavallar, S.V.; Yohanandan, S.; Roy, S.; Tang, J.; Harrer, S. Privacy Preserving Human Fall Detection Using Video Data. In Proceedings of the Machine Learning for Health NeurIPS Workshop (PMLR), PMLR 116, Virtual, 11 December 2020; pp. 39–51.
17. Wang, R.D.; Zhang, Y.L.; Dong, L.P.; Lu, J.W.; Zhang, Z.Q.; He, X. Fall detection algorithm for the elderly based on human characteristic matrix and SVM. In Proceedings of the 2015 15th International Conference on Control, Automation and Systems (ICCAS), Busan, Republic of Korea, 13–16 October 2015; pp. 1190–1195.
18. Liu, L.; Hou, Y.; He, J.; Lungu, J.; Dong, R. An Energy-Efficient Fall Detection Method Based on FD-DNN for Elderly People. *Sensors* **2020**, *20*, 4192. [CrossRef]
19. Nahian, M.J.A.; Ghosh, T.; Banna, H.A.; Aseeri, M.A.; Uddin, M.N.; Ahmed, M.R.; Mahmud, M.; Kaiser, M.S. Towards an Accelerometer-Based Elderly Fall Detection System Using Cross-Disciplinary Time Series Features. *IEEE Access* **2021**, *9*, 39413–39431. [CrossRef]
20. Lin, C.-L.; Chiu, W.-C.; Chu, T.-C.; Ho, Y.-H.; Chen, F.-H.; Hsu, C.-C.; Hsieh, P.-H.; Chen, C.-H.; Lin, C.-C.K.; Sung, P.-S.; et al. Innovative Head-Mounted System Based on Inertial Sensors and Magnetometer for Detecting Falling Movements. *Sensors* **2020**, *20*, 5774. [CrossRef] [PubMed]

21. Fall Detection. Available online: https://www.lively.com/services-apps/fall-detection/ (accessed on 3 July 2024).
22. Protection in and Away from the Home. Available online: https://www.bayalarmmedical.com/medical-alert-system/bundle/ (accessed on 3 July 2024).
23. Use Fall Detection with Apple Watch. Available online: https://support.apple.com/en-us/HT208944 (accessed on 3 July 2024).
24. Ngu, A.; Wu, Y.; Zare, H.; Polican, A.; Yarbrough, B.; Yao, L. Fall detection using smartwatch sensor data with accessor architecture. In Proceedings of Smart Health: International Conference (ICSH), Hong Kong, China, 26–27 June 2017; pp. 81–93.
25. Wu, X.; Zheng, Y.; Chu, C.H.; Cheng, L.; Kim, J. Applying deep learning technology for automatic fall detection using mobile sensors. *Biomed. Signal Process. Control.* **2022**, *72*, 103355. [CrossRef]
26. Şengül, G.; Karakaya, M.; Misra, S.; Abayomi-Alli, O.O.; Damaševičius, R. Deep learning based fall detection using smartwatches for healthcare applications. *Biomed. Signal Process. Control.* **2022**, *71*, 103242. [CrossRef]
27. Mauldin, T.R.; Canby, M.E.; Metsis, V.; Ngu, A.H.; Rivera, C.C. SmartFall: A smartwatch-based fall detection system using deep learning. *Sensors* **2018**, *18*, 3363. [CrossRef]
28. Fitbit Sense. Available online: https://www.fitbit.com/global/us/products/smartwatches/sense (accessed on 3 July 2024).
29. Samsung Galaxy A32. Available online: https://www.samsung.com/it/smartphones/galaxy-a/galaxy-a32-5g-white-128gb-sm-a326bzwveue/ (accessed on 3 July 2024).
30. Bozinovski, S. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica* **2020**, *44*, 17. [CrossRef]
31. Najmi, A.H.; Sadowsky, J. The continuous wavelet transform and variable resolution time-frequency analysis. *Johns Hopkins Apl Tech. Dig.* **1997**, *18*, 134–140.
32. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]
33. WEDA-FALL. Available online: https://github.com/joaojtmarques/WEDA-FALL (accessed on 3 July 2024).
34. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Spinger: Berlin/Heidelberg, Germany, 2009.
35. Marques, J.; Moreno, P. Online fall detection using wrist devices. *Sensors* **2023**, *23*, 1146. [CrossRef]
36. Kim, J.K.; Lee, K.; Hong, S.G. Detection of important features and comparison of datasets for fall detection based on wrist-wearable devices. *Expert Syst. Appl.* **2023**, *234*, 121034. [CrossRef]
37. Vavoulas, G.; Pediaditis, M.; Chatzaki, C.; Spanakis, E.G.; Tsiknakis M. The Mobifall Dataset: Fall detection and classification with a smartphone. *Int. J. Monit. Surveill. Technol. Res. (IJMSTR)* **2014**, *2*, 44–56. [CrossRef]
38. Sucerquia, A.; López, J.D.; Vargas-Bonilla, J.F. SisFall: A Fall and Movement Dataset. *Sensors* **2017**, 17, 198. [CrossRef]
39. Medrano, C.; Igual, R.; Plaza, I.; Castro, M. Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE* **2014**, *9*, e94811. [CrossRef]