mathematics

MDPI

Article

S3D: Squeeze and Excitation 3D Convolutional Neural Networks for a Fall Detection System

Seung Baek Hong, Yu Hwan Kim, Se Hyun Nam and Kang Ryoung Park *

Division of Electronics and Electrical Engineering, Dongguk University, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Korea; baek2sm@dgu.ac.kr (S.B.H.); taekkuon@naver.com (Y.H.K.); nsh6473@naver.com (S.H.N.)
* Correspondence: parkgr@dongguk.edu; Tel.: +82-10-3111-7022; Fax: +82-2-2277-8735

Abstract: Because of the limitations of previous studies on a fall detection system (FDS) based on wearable and ambient devices and visible light and depth cameras, the research using thermal cameras has recently been conducted. However, they also have the problem of deteriorating the accuracy of FDS depending on various environmental changes. Given these facts, in this study, we newly propose an FDS method based on the squeeze and excitation (SE) 3D convolutional neural networks (S3D). In our method, keyframes are extracted from input thermal videos using the optical flow vectors, and the fall detection is carried out based on the output of the proposed S3D, using the extracted keyframes as input. Comparative experiments were carried out on three open databases of thermal videos with different image resolutions, and our proposed method obtained F1 scores of 97.14%, 95.30%, and 98.89% in the Thermal Simulated Fall, Telerobotics and Control Lab fall detection, and eHomeSeniors datasets, respectively (the F1 score is a harmonic mean of recall and precision; it was confirmed that these are superior results to those obtained using the state-of-the-art methods of a thermal camera-based FDS.

Keywords: fall detection system; thermal video; deep learning; squeeze and excitation; 3D CNN

Citation: Hong, S.B.; Kim, Y.H.; Nam, S.H.; Park, K.R. S3D: Squeeze and Excitation 3D Convolutional Neural Networks for a Fall Detection System. *Mathematics* **2022**, *10*, 328. https://doi.org/10.3390/math10030328

Academic Editors: George E. Tsekouras, Christos Kalloniatis and Dimitrios Makris

Received: 3 January 2022
Accepted: 20 January 2022
Published: 21 January 2022

1. Introduction

With the progress of the aging population across the globe, the number of falling accidents is increasing, and the injury rate in the case of falling accidents has reached 20–30%. Moreover, more than half of people injured and hospitalized owing to falling accidents are the elderly, aged over 65 [1]. The elderly cannot easily stand up by themselves in the case of a falling accident because of deteriorating muscle function, and if they remain on the floor for an extended period after a falling accident, they risk suffering from dehydration or hypothermia, which can be fatal in some cases; thus, they are more vulnerable to safety risks attributed to falling accidents [2].

A fall detection system (FDS) can be an alternative solution to mitigate the elderly's safety problem as aforementioned. By automatically detecting falling accidents of the elderly and informing an appropriate organization of the dangerous situation, the FDS can prevent the elderly from running into more severe danger due to falling accidents. In recent years, with the advancements of basic technology, including the internet and artificial intelligence, research on FDSs has been actively performed [3]. In the case of existing FDS methods using wearable devices, their wide utilization is impeded by the disadvantage that the device must be attached to the body or be carried at all times. On the contrary, ambient device-based FDSs that operate in a sensor-embedded environment, such that the sensor does not need to be attached to the body, are more convenient. However, systems that use several sensors suffer from the disadvantage of sensitivity to the location or angle of the sensors [4]. To this end, vision-based FDS studies have been conducted, and vision-based FDSs have the advantage of less sensitivity to the location or angle of the sensor compared with FDSs that require the accurate placement of several sensors. However, visible light

cameras that are widely used in vision-based systems pose concerns of privacy invasion, and in terms of depth cameras, their fall detection accuracy decreases in the presence of other objects near the target. For these reasons, FDS studies using thermal cameras that can identify a person's shape even at night have been carried out in recent years. Fall detection belongs to the action recognition field, and in this field, existing 3-dimensional (3D) convolutional neural networks (CNNs) have shown excellent performance [5,6]. Given these facts, in this study, we newly propose an FDS method based on the squeeze and excitation (SE) 3D CNN (S3D), which can improve the accuracy of the fall detection by applying a revised SE block considering 3D CNN. This study makes contributions in the following four aspects compared with the existing studies.

- This is the first study on FDSs in which an SE block is combined with a 3D CNN. The S3D-based FDS proposed in this study showed a higher detection performance than the state-of-the-art methods on open datasets with various resolutions.
- Fall detection was carried out by extracting the keyframes from the input thermal video using the magnitudes of the optical flow vectors and using the extracted keyframes as the input of the S3D.
- The SE block previously used only in a 2-dimensional (2D) CNN was newly transformed to be suitable for a 3D CNN and applied. Furthermore, the effect of the SE block location in the 3D CNN model on the network was analyzed through gradient-weighted class activation mapping (Grad-CAM), and it demonstrated where the SE block should be located and how many SE blocks should be used to obtain the best performance.
- For other researchers to compare and evaluate its performance, the S3D model used in this study is made public through [7].

The remainder of this paper is organized as follows. In Section 2, the related works are discussed, and in Section 3, our proposed method is explained. In Section 4, the experimental results and discussion are provided. Finally, in Section 5, the conclusions of this study are summarized, and the potential directions for future research direction are discussed.

2. Related Works

The existing studies on FDSs can mainly be classified into those on wearable device-based, ambient device-based, and vision-based systems.

2.1. Wearable Device-Based Methods

Typical wearable device-based systems use an accelerometer, possibly in combination with a gyroscope. Because acceleration or gradient rapidly change in the case of falling accidents, falls can be detected using the accelerometer and gyroscope attached to the human body.

2.1.1. Accelerometer-Based

In previous research [8], a fall was recognized if the acceleration change detected by the accelerometer exceeded a certain threshold value. Although the calculation volume is very low, and hence, a low-specification processor can be utilized, it does not take into account the movement direction and only considers its magnitude, leading to a high probability of false detections when there is any dynamic movement. In an existing study [9], falls were detected according to threshold values obtained through a decision tree and state machine. In this method, the threshold value was determined by the decision tree; thus, if sufficient data were provided, higher performance could be expected compared with threshold values chosen arbitrarily by a human. However, it is difficult to detect all fall situations using a fixed threshold value, and there is a limitation that the judgment is made depending on the learned threshold value.

2.1.2. Fusing Accelerometer and Gyroscope-Based

In previous studies [10,11], falls were recognized by using an accelerometer and gyroscope, setting the threshold values of the acceleration and angle changes, and detecting whether these two readings exceeded the threshold values. Although there are the advantages that the fall detection can be performed using very low-specification processors, and it is more accurate compared with using only one sensor, it still considers the instantaneous change, and thus, it may not detect slower falling movements.

Although not about the research of FDS, Pourbemany et al. proposed Breath to Pair (B2P), a protocol for pairing and shared-key generation for wearable devices that uses the wearer's respiration activity to ensure that the devices become part of the same body-area network [12]. In addition, authors surveyed context-based pairing in wearable devices by focusing on the signals and sensors exploited, and they reviewed the steps needed for generating a common key and provided a survey of existing techniques used in each step [13].

2.2. Ambient Device-Based Methods

Such wearable device-based systems [8–11] are commonly preferred with respect to privacy protection but have the shortcoming that the sensor must be directly attached to the body or be carried at all times. As an alternative, studies using an ambient device typically utilize radar and ultrasonic sensors.

2.2.1. Radar-Based

In a previous study [14], a radar sensor was used to detect falls through an autoencoder and logistic regression. Although it has the advantage of fewer concerns about privacy invasion compared with vision-based methods, it has the disadvantage of requiring an expensive sensor.

2.2.2. Ultrasonic-Based

In the existing research, fall detection was carried out through several ultrasonic sensors and event pattern matching. Although it is preferred that they are utilized in bathrooms and toilets because the sensor does not need to be directly attached to the body, and there are fewer concerns about privacy invasion of an individual compared with cameras, it has the disadvantage of increased misclassification probability in the presence of other moving objects, such as pets or robotic vacuums, as the sensor is installed at an altitude close to the floor [15].

Although it is not about the research of FDS, Sanaat et al. proposed a supervised deep neural network which was used for the approximation of the depth of interaction (DOI) and to evaluate, through Monte Carlo (MC) simulations, the performance on a small-animal positron emission tomography (PET) scanner [16].

2.3. Vision-Based Methods

Studies on vision-based methods are classified into those using visible light cameras, those using depth cameras, and those using thermal cameras.

2.3.1. Visible Light Camera-Based

The previous study [17] used a visible light camera and recognized falls with the K-nearest neighborhood algorithm (KNN) by applying foreground extraction to the video and obtaining the optical flow vector value for the foreground area. Because the visible light camera is employed, the system can be established at a low cost. However, the visible light camera is extremely vulnerable to the privacy problem.

2.3.2. Depth Camera-Based

In a previous study [18], the region of interest (ROI) was detected using the depth camera, and the fall situation was recognized based on the cases in which the changes

of pixel positions of the ROI region were larger than a threshold value. This method has the advantage that it can generate a system without much data, as it is not a training-based method; however, it does not take into account the speed or direction of a human's movement, leading to errors in the actual environment. In another study [19], the human outline was extracted from the video with the canny filter using the depth camera, and the fall was recognized according to the tangential distribution of the outline pixels. Because the depth camera is utilized, this method can identify a person even in the dark; however, its accuracy can be reduced if there are many other objects nearby.

2.3.3. Thermal Camera-Based

Using a thermal camera, a previous study [20] recognized the fall with the support vector machine (SVM), utilizing x-axis and y-axis histograms calculated from the image as a feature. Although this method has fewer features used in the SVM, and thus, its calculation speed is fast, the pixel distribution changes if the environment differs from that in which the training data were obtained, and the accuracy cannot be guaranteed. In a previous study [21] that employed a thermal camera, a frequency analysis was conducted on the angle of the optical flow vector with the fast Fourier transform (FFT), and the fall detection was performed with the SVM. This method has the advantage that the accuracy change is not significant in different environments; however, its fall detection accuracy is generally low. The existing studies [22,23] utilized an autoencoder, which encoded the thermal video with the convolutional long short-term memory (ConvLSTM) and then decoded it, and the fall detection was carried out by comparing the difference between the original video and the reconstructed video. This method is based on a recurrent neural network and can extract the temporal information from the continuous frames; however, in the case of a movement pattern that was not learned with the training dataset, the risk of misclassification increases. In a previous study [24], falls were detected with a 3D CNN by extracting keyframes from a thermal video based on the magnitudes of the optical flow vectors. This method is effective as it only uses the frames with the most information as the network input, but it has the shortcoming of deteriorating accuracy under various environmental changes.

Although not about the research of FDS, Bangtal et al. proposed a new variant of the Bat algorithm (BA), named as the improved bat algorithm (IBA), which modifies the standard BA by enhancing its exploitation capabilities and avoids escaping from local minima [25]. In other research [26], the authors proposed a new initialization population approach, termed as the enhanced version of particle swarm optimization (PSO) following Log-logistic distribution as Log-logistic Neural Network (PSOLL-NN), to create the initialization of the swarm. In addition, Castellano et al. proposed a crowd detection method for drone safe landing based on the light–weight scheme of a fully convolutional neural network which conjugates effectiveness and nimble computations [27], and it can be considered for the faster computation of our method.

Although not about the research of FDS, in [28], the authors adapted the SE block to 3D CNNs for the micro expression recognition of a face image. In [29], the authors also combined the SE block and 3D CNNs for remote sensing sea ice image classification. Although the concept of applying the SE block to a 3D CNN in [28,29] is similar to that of our research, the structure of our final model (S3D), combining the SE block and 3D CNN, is completely different from those of their methods [28,29]. In addition, they did not analyze the effect of the SE block on the 3D CNN according to the location of the SE blocks. However, we thoroughly analyzed not only the results according to the location of SE blocks, but also why the model learned using SE blocks showed better results by visualizing the activation intensity for each channel of SE blocks.

Given these problems, our proposed method extracts keyframes using the magnitudes of the optical flow vectors and detects falls with the 3D CNN with a SE block applied. By applying the SE block to one of the 3D CNN networks, namely convolutional 3D (C3D) [5],

and considering the important spatiotemporal information, the fall detection is carried out accurately.

3. Proposed Method

3.1. Overview of the Proposed Architecture

Figure 1 shows the overall flowchart of the proposed method. The method first extracts keyframes based on the magnitudes of the optical flow vectors extracted from the input thermal image sequences (steps (1) and (2) of Figure 1). The processing time can be reduced by using the frames with the largest changes in the optical flow vector values in two consecutive frames as keyframes, without losing important information while not processing all the frames. Afterward, the extracted keyframes are subjected to size normalization into the size of 112 × 112 pixels (steps (3) of Figure 1), and the 3D input of dimensions $N \times 112 \times 112$ is obtained through the depth-wise composition with the extracted N keyframes (steps (4) of Figure 1). The obtained 3D input is passed to the proposed network as input (step (5) of Figure 1). In this study, while using the C3D, a 3D CNN model, as a backbone, the SE block is added to the C3D to improve the network performance through the feature recalibration, and the newly proposed S3D model is adopted. The SE block can improve the performance while not significantly increasing the number of weights by using the principle of giving attention to the channel with more important information in the feature map. Subsequently, the fall detection is conducted based on the output values of S3D (step (6) of Figure 1). During fall detection, the classification of two classes of fall and activities of daily living (ADL, not fall) is performed. In the next subsections, this proposed method is explained in detail.

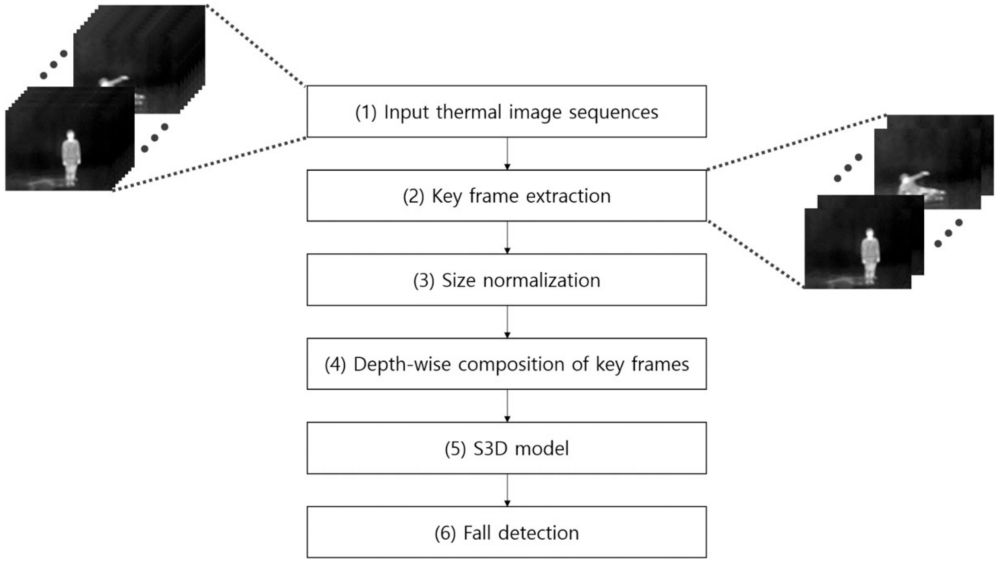


Figure 1. Overview of the proposed method.

3.2. Keyframe Extraction and Depth-Wise Composition of Extracted Keyframes

In this study, the keyframes of the input thermal image sequences are used as the network input. To extract the keyframes, the optical flow vectors are calculated in two adjacent frames, and the frames with the largest range of the optical flow vectors are used as keyframes in the sequence. When calculating the optical flow vectors, the Lucas–Kanade–Tomasi method [30] is utilized, in which the corner points to track are extracted through the Shi–Tomasi corner detection [31], and then, the optical flow vectors are calculated using the Lucas–Kanade method [32]. First, for extracting the feature points to track, the structure tensor Z is derived in the front frame image between two adjacent frames using the Taylor expansion [31], and the minimum of the eigenvalues of Z, namely μ_1 and μ_2, is used as the corner response score V. Next, if V is larger than the threshold value, it is determined

to be the corner to track. The V value can be calculated using the following equation [33]. In Equation (1), *det* represents determinant, and *tr* represents trace, and in this study, the optimal k value was experimentally determined as the value that results in the best accuracy of fall detection using the training data.

$$V = det(Z) - k(tr(Z))^2 \tag{1}$$

Using the V value, the feature points to track are determined in the front frame between two adjacent frames, and the optical flow vectors are calculated based on these feature points. The calculation of the optical flow vectors assumes that the pixel intensities of an object between two adjacent frames do not change and that the adjacent pixels within the frame have a similar motion. Therefore, the following equation is valid when a pixel, $P(x,\ y,\ t)$, moves by $\alpha x,\ \alpha y$ after time αt (t represents time):

$$P(x,\ y,\ t) = P(x + \alpha x,\ y + \alpha y,\ t + \alpha t) \tag{2}$$

Subsequently, the following optical flow equation is derived after taking the Taylor series approximation on the right side of Equation (2), canceling out the common terms, and dividing both sides by αt [34]:

$$f_x n + f_y m + f_t = 0 \tag{3}$$

where

$$f_x = \frac{\beta f}{\beta x};\ f_y = \frac{\beta f}{\beta y}$$
$$n = \frac{\alpha x}{\alpha t};\ m = \frac{\alpha y}{\alpha t}$$

In Equation (3), f_x and f_y signify the gradients of the image, and f_t means the time gradient, and when the number of feature points to track is N, n and m are calculated with the following equations [34] using the Lucas–Kanade method [32]:

$$\begin{bmatrix} n \\ m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} {f_{x_i}}^2 & \sum_{i=1}^{N} f_{x_i} f_{y_i} \\ \sum_{i=1}^{N} f_{x_i} f_{y_i} & \sum_{i=1}^{N} {f_{y_i}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{N} f_{x_i} f_{t_i} \\ -\sum_{i=1}^{N} f_{y_i} f_{t_i} \end{bmatrix} \tag{4}$$

In this study, for keyframes, 16 frames with the largest sum of the magnitudes of optical flow vectors calculated in the two adjacent frames were selected, among all frames of the input thermal image sequences. During this process, between two adjacent frames, the altered frame was used as a keyframe, and in this study, the optimal number of keyframes (16) was experimentally determined as a value resulting in the best accuracy of fall detection using the training data. The 16 extracted keyframes were subjected to size normalization to the size of 112 × 112 pixels by bilinear interpolation, and the 3D input of dimensions 16 × 112 × 112 were obtained using the extracted keyframes through the depth-wise composition. The obtained 3D input was used as the input of the S3D to be explained in the next subsection.

3.3. Structure of the Proposed S3D Model

The proposed S3D model extends the SE block [35] to 3D, focuses on the channel with more informative features in the 3D features map, and improves the FDS performance. In this study, the C3D model [5] was used as a 3D CNN backbone model, and the S3D model was proposed, in which the SE block was added immediately after the last max pooling layer of the C3D model, as presented in Figure 2. Through the ablation studies presented in Sections 4.3.1–4.3.3, it was identified that arranging the SE block behind the last max pooling layer of the C3D model results in the best performance.

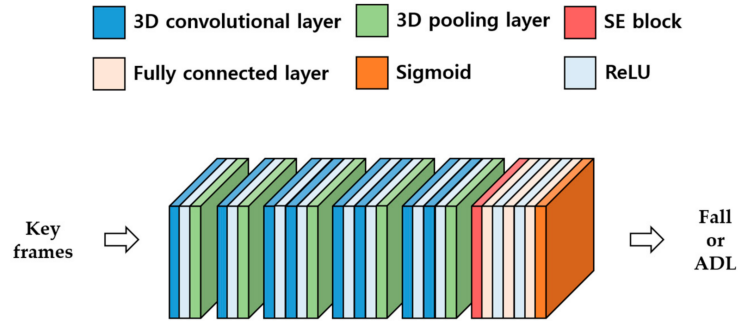


Figure 2. Proposed S3D model.

As presented in Figure 2, the architecture of the S3D model is composed of eight 3D convolutional layers, five 3D max pooling layers, one SE block, and three fully connected (FC) layers, and apart from the last FC layer, in which the sigmoid function was used, the rectified linear unit (ReLU) [36] was used after all convolutional layers and FC layers. According to the literature [5], 3D CNN yields the best performance when the size of the convolutional filter is $3 \times 3 \times 3$; hence, in this study, all 3D convolutional filters were of dimensions $3 \times 3 \times 3$, and the filters for the 3D max pooling were of dimensions $2 \times 2 \times 2$. However, to prevent early loss of temporal information of the input data, the filter with the depth of 1, i.e., with dimensions $1 \times 2 \times 2$, was exceptionally used in the first max pooling layer. The SE block was used immediately after the last max pooling layer, and in the SE block, two FC layers, the ReLU function, and the sigmoid function were utilized for the feature recalibration. After conducting the feature recalibration through the SE block, the final prediction was carried out using two FC layers and one sigmoid output layer. Table 1 shows the detailed architecture of the proposed model.

Table 1. S3D model architecture (the Rectified linear unit (ReLU) activation function was used behind all 3D convolutional layers).

Layer Name	Size of Filter (Depth × Height × Width)	Number of Filters	Stride (Depth × Height × Width)	Padding (Depth × Height × Width)	Size of the Feature Map (Depth × Height × Width)
3D convolutional layer	3 × 3 × 3	64	1 × 1 × 1	1 × 1 × 1	16 × 112 × 112
3D pooling layer	1 × 2 × 2	-	1 × 2 × 2	-	16 × 56 × 56
3D convolutional layer	3 × 3 × 3	128	1 × 1 × 1	1 × 1 × 1	16 × 56 × 56
3D pooling layer	2 × 2 × 2	-	2 × 2 × 2	-	8 × 28 × 28
3D convolutional layer	3 × 3 × 3	256	1 × 1 × 1	1 × 1 × 1	8 × 28 × 28
3D convolutional layer	3 × 3 × 3	256	1 × 1 × 1	1 × 1 × 1	8 × 28 × 28
3D pooling layer	2 × 2 × 2	-	2 × 2 × 2	-	4 × 14 × 14
3D convolutional layer	3 × 3 × 3	512	1 × 1 × 1	1 × 1 × 1	4 × 14 × 14
3D convolutional layer	3 × 3 × 3	512	1 × 1 × 1	1 × 1 × 1	4 × 14 × 14
3D pooling layer	2 × 2 × 2	-	2 × 2 × 2	-	2 × 7 × 7
3D convolutional layer	3 × 3 × 3	512	1 × 1 × 1	1 × 1 × 1	2 × 7 × 7
3D convolutional layer	3 × 3 × 3	512	1 × 1 × 1	1 × 1 × 1	2 × 7 × 7
3D pooling layer	2 × 2 × 2	-	2 × 2 × 2	0 × 1 × 0	1 × 4 × 3
3D SE block: FC	-	-	-	-	32
3D SE block: ReLU	-	-	-	-	32
3D SE block: FC	-	-	-	-	512
3D SE block: sigmoid	-	-	-	-	512
FC	-	-	-	-	4096
ReLU	-	-	-	-	4096
FC	-	-	-	-	4096
ReLU	-	-	-	-	4096
FC	-	-	-	-	1
sigmoid	-	-	-	-	1

The SE block carries out the feature recalibration through the squeeze and excitation operations. In the squeeze operation, to obtain the channel-wise global information statics, the $1 \times 1 \times 1 \times C$-sized feature map is produced while maintaining the number of channels, C, constant through the 3D global average pooling. In other words, the squeeze operation, in which the cth channel of the 3D feature map U, i.e., U_c, is converted to the global information statics, s_c, though the 3D global average pooling, is calculated using the following equation (D, H, and W represent depth, height, and width, respectively, and here, the depth refers to the frame axis.):

$$s_c = \frac{1}{D \times H \times W} \sum_{i=1}^{D} \sum_{j=1}^{H} \sum_{k=1}^{W} U_c(i,j,k) \tag{5}$$

The statics value obtained through the squeeze operation, s_c, can be regarded as a result of compressing the global spatiotemporal information by channel. Next, in the excitation operation, two FC layers, the ReLU function, and the sigmoid function are used to identify the channel-wise dependencies. During this process, to reduce the complexity of the model, the bottleneck is derived by decreasing the number of nodes in the first FC layer to $\frac{1}{r}$. It was experimentally verified in the existing study [35] that removing biases in the FC layer is effective for analyzing channel-wise dependencies and that the r value for the suitable balance between the model complexity and accuracy is 16. Thus, in this study, the biases were removed from the FC layer, and the reduction ratio r was set to a value of 16. The equation for calculating the value, b, that derives the bottleneck in the global information statics, s, though the first FC layer is as follows (P signifies the trainable parameters of the first FC layer, and ω signifies the ReLU operation):

$$b = \omega \left(\begin{bmatrix} P_{1,1} & \cdots & P_{1,c} \\ \vdots & \ddots & \vdots \\ P_{\frac{c}{r},1} & \cdots & P_{\frac{c}{r},c} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{c-1} \\ s_c \end{bmatrix} \right) \tag{6}$$

After deriving the bottleneck through the first FC layer, the data length is set to be the same as the number of channels through the second FC layer, and the data values are set to be between 0 and 1 through the sigmoid function. Via this process, the channel-wise dependencies can be identified, and the value for the activation of more important channels, i.e., e, can be obtained. The equation for this process is as follows (p means the parameters of the second FC layer, and μ means the sigmoid operation):

$$e = \mu \left(\begin{bmatrix} p_{1,1} & \cdots & p_{1,\frac{c}{r}} \\ \vdots & \ddots & \vdots \\ p_{c,1} & \cdots & p_{c,\frac{c}{r}} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{\frac{c}{r}-1} \\ b_{\frac{c}{r}} \end{bmatrix} \right) \tag{7}$$

Lastly, through the scale operation that multiplies e by the feature map, U, channel-wise, the recalibrated feature map, U', can be obtained by emphasizing the channels with more important information. One channel of the recalibrated feature map U', namely U'_c, can be obtained by multiplying the scalar value, s_c, by the 3D vector value of one channel of U, U_c, and the equation for this process is as follows:

$$U'_c = s_c U_c \tag{8}$$

Figure 3 shows the squeeze, excitation, and scale operations of the SE block that recalibrate the feature map channel-wise, depending on the importance of each channel.

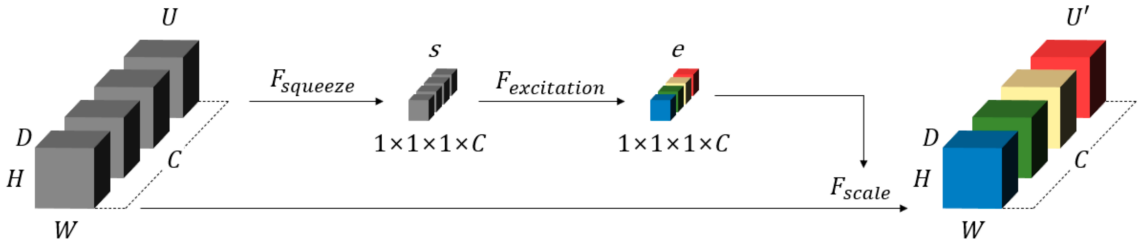


Figure 3. 3D squeeze and excitation block. $F_{squeeze}$ is the squeeze operation, $F_{excitation}$ is the excitation operation, and F_{scale} is the scale operation.

Figure 4 presents a comparison between the diagrams of the 3D convolutional layer and the 3D convolutional layer with the SE block proposed in this study. As shown in Figure 4, the 3D convolutional layer simply utilizes 3D convolutional filters, while the 3D convolutional layer with the SE block emphasizes more important channels through squeeze and excitation operations.

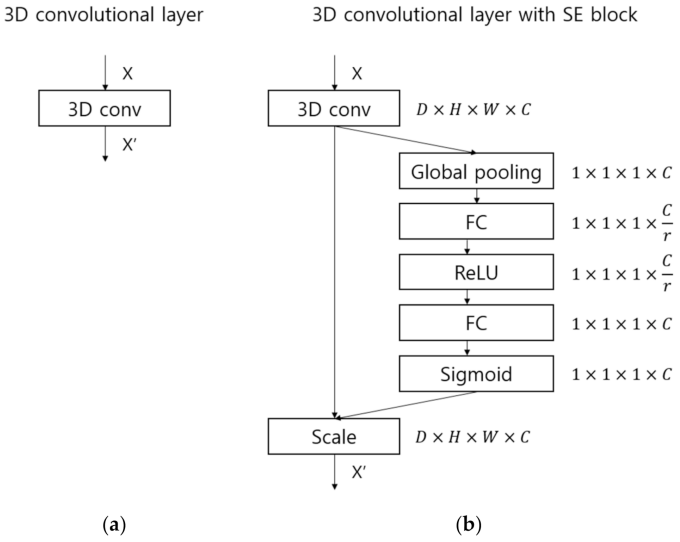


Figure 4. Comparative diagrams of (**a**) the 3D convolutional layer and (**b**) the 3D convolutional layer with the SE block.

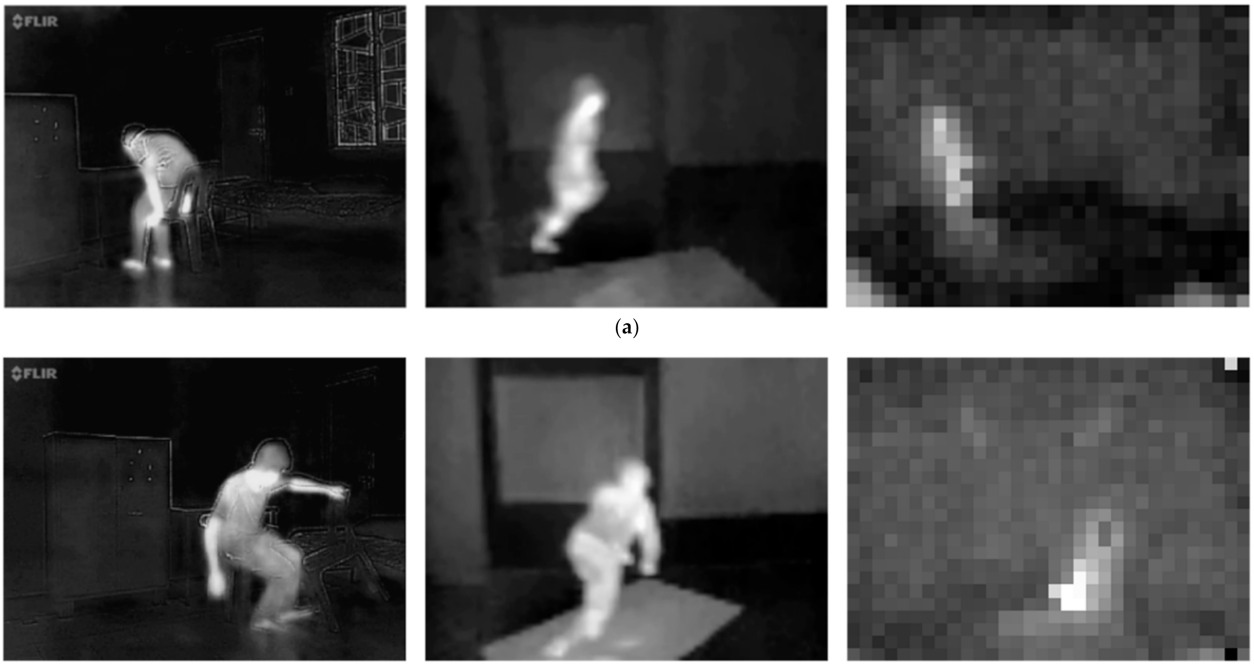
3.4. Differences between the Proposed S3D and Previous Methods

- In the study [35] that proposes the SE block, the feature recalibration was carried out considering only spatial dimensions (height × width), as it was only focused on the 2D feature map. However, in this study, the feature recalibration was conducted considering the spatiotemporal dimensions (depth × height × width), as it is focused on the 3D feature map.
- In the study [5] that proposes the C3D, frames in several sections were uniformly selected and used as the model input; thus, the model was repeatedly utilized in the inference process. However, in this study, the model was used only once in the inference process by selecting only keyframes and using them as the model input. Moreover, by adding the SE block into the structure of the C3D model, the S3D model that considers the information of the more important channel was newly proposed in this study.
- In the study [24], the 3D CNN-based fall detection was performed. However, in this study, a suitable structure of the 3D CNN model for an FDS was sought by utilizing the SE block, and the S3D model-based FDS that is robust to environmental changes was newly proposed.

4. Experiments

4.1. Datasets and Experimental Environments

For the performance verification of the proposed method, the model performance was compared using three fall detection open datasets composed of thermal videos. The three datasets used in this study were the Thermal Simulated Fall dataset (TSF dataset) [21], the Telerobotics and Control Lab fall detection dataset (TCL dataset) [24], and the eHomeSenior dataset [37]. The TSF dataset consists of a total of 44 videos, including 35 fall videos with a 640 × 480 pixels resolution and 9 ADL videos. The TCL dataset is composed of a total of 1252 videos, including 421 fall videos with a 640 × 480 pixels resolution and 831 ADL videos. The eHomeSenior dataset consists of one dataset with a 1 × 8 pixels resolution and another with a 32 × 24 pixels resolution, and we chose to use the dataset with a 32 × 24 pixels resolution. A total of 448 fall accidents are included in the eHomeSenior dataset. Similar to existing studies, the frames corresponding to the fall were annotated from the videos, the other parts were regarded as ADL, and finally, they were classified into a total of 921 videos, including 448 fall videos and 473 ADL videos [22]. Sample images of the dataset used in this study are shown in Figure 5, and the composition of the dataset is presented in Table 2. In addition, the detailed explanations of percentages of dataset used for training and testing are shown in Table 2. For example, in case of the eHomeSenior dataset, 322 (157 + 165) (35%), 138 (67 + 71) (15%), and 461 (224 + 237) (50%) videos among a total of 921 (448 + 473) videos were used for training, validation, and testing, respectively.



(**a**)

(**b**)

Figure 5. Sample images of the TSF, TCL, and eHomeSenior datasets from the left. (**a**) Represents the sample images of ADL videos, and (**b**) represents the sample images of fall videos.

The proposed method was implemented using OpenCV version 4.5.3 [38] and Pytorch 1.7.1 [39] in the Ubuntu 18.04 operating system (OS), NVIDIA compute unified device architecture (CUDA) version 11.0 [40], and NVIDIA CUDA® deep neural network library (CUDNN) version 8.0 [41]. The desktop computer used for the experiments included an Intel® Core-i7-4770 central processing unit (CPU), 16 GB random access memory (RAM), and an NVIDIA GeForce GTX Titan X graphic processing unit (GPU) [42].

Table 2. Detailed information on datasets used in the comparative experiments of this study.

Dataset	Resolution (Width × Height) (Unit: Pixels)	Number of Fall Videos (Training/Validation/Testing)	Number of ADL Videos (Training/Validation/Testing)	Total
TSF	640 × 480	35 (12/5/18)	9 (3/1/5)	44
TCL	640 × 480	421 (147/63/211)	831 (290/125/416)	1252
eHomeSenior	32 × 24	448 (157/67/224)	473 (165/71/237)	921

4.2. Training

To reduce the processing time and complexity of the S3D model, the TSF and TCL datasets were resized to the resolution of 112 × 112 pixels by bilinear interpolation. In terms of the eHomeSenior dataset, the resolution of the original video was significantly lower than this, as presented in Table 2; hence, the 32 × 24 pixels resolution was not modified. All experiments were carried out using two-fold cross validation. To this end, as presented in Table 2, approximately half of the fall and ADL videos in the TSF, TCL, and eHomeSenior datasets were used as training data in the 1st fold validation, and the remaining videos were used as testing data. Furthermore, 30% of videos of the training data were utilized as validation data. Afterward, in the 2nd fold validation, the training and testing data were swapped; the training, validation, and testing were conducted once more; and the average performance of these two tests was determined to be the final testing accuracy. Moreover, given the open world configuration, videos of the same scene were not included in both training and testing data. Table 3 shows the hyperparameters used for the training of our model.

Table 3. The hyperparameters of our proposed model.

Weight Decay	Loss	Kernel Initializer	Bias Initializer	Optimizer	LEARNING RATE	Beta_1	Beta_2	Epsilon	Batch Size
0.5	"binary cross-entropy loss"	"He uniform"	"zeros"	"adam"	0.0001	0.9	0.999	1×10^{-8}	16

As a loss function of the proposed S3D model, the binary cross entropy loss [43] was used. The kernels of the network were initialized to 'He uniform' [44], and the bias was initialized to zero. For the model learning, the batch size was set at 16, and the adaptive moment estimation (Adam) optimizer [45] was utilized. The beta values of the Adam optimizer are 0.9 and 0.999, respectively, and the epsilon value is 1×10^{-8}. For the model learning, the batch size was set at 16, and the adaptive moment estimation (Adam) optimizer was utilized. The training was carried out at the learning rate of 0.0001, while at every 16th epoch, the learning rate decayed by half, and learning was performed. Figure 6 shows the training loss and validation loss graphs of the proposed method. As shown in Figure 6, the training loss graphs converged with an increasing number of iterations, and based on these results, it was identified that the proposed S3D model showed sufficient learning about the training data. Furthermore, the validation loss graphs in Figure 6 also converged with the increase of the number of iterations, and these results indicate that the S3D model was not overfitted to the training data.

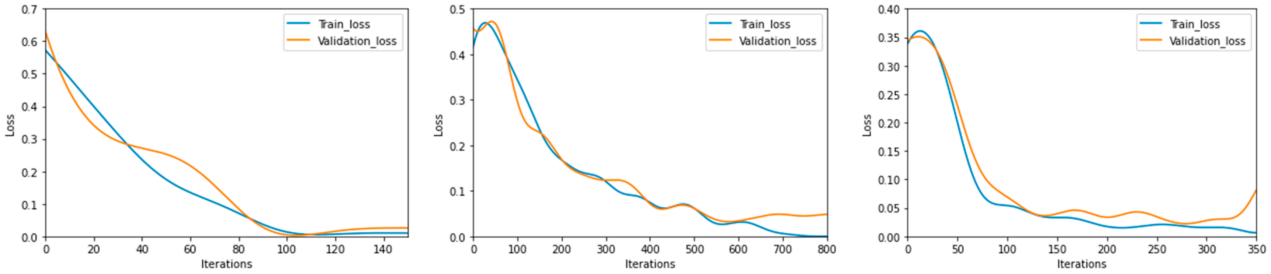


Figure 6. Training loss and validation loss graphs of the S3D model with the (**a**) TSF dataset, (**b**) TCL dataset, and (**c**) eHomeSenior dataset.

4.3. Experimental Results

The proposed models, including state-of-the-art models, were evaluated in terms of accuracy, F1 score, Precision, and Recall. In addition, the assessment matrices are defined as in [46].

$$\text{Accuracy} = \frac{\#\text{ of } (\text{TP} + \text{TN})}{\#\text{ of } (\text{TP} + \text{FN} + \text{FP} + \text{TN})} \tag{9}$$

$$\text{Recall} = \frac{\#\text{ of } (\text{TP})}{\#\text{ of } (\text{TP} + \text{FN})} \tag{10}$$

$$\text{Precision} = \frac{\#\text{ of } (\text{TP})}{\#\text{ of } (\text{TP} + \text{FP})} \tag{11}$$

$$\text{F1 score} = \frac{2\text{RecallPrecision}}{\text{Recall} + \text{Precision}} \tag{12}$$

where TP, TN, FP, and FN represent the true positive, true negative, false positive, and false negative, respectively. In particular, TP and TN are the correctly predicted positive (fall class) and negative (no-fall class) cases by our proposed network, whereas FP and FN are the incorrectly predicted positive and negative cases, respectively. '#' means 'the number of'.

4.3.1. TSF Dataset

Ablation Studies

Table 4 presents the ablation study results on using keyframes as the network input in the C3D, a 3D CNN model that is the foundation of the S3D model proposed in this study, and adding the SE block to the network. In the case of not using keyframes, the video clips at a fixed length were randomly extracted from the videos during the training step, and 10 video clips at a fixed length were extracted by a constant interval, and the average value of sigmoid scores was used for the prediction in the inference step, similar to the methods used in previous studies [6,47]. As presented in Table 4, the F1 score of the case in which the keyframes were used as the network input in the C3D model was 95.77%, whereas that of the case where the SE block was added to the C3D model was 95.65%. Compared with the F1 score of the case using only the C3D model, 94.29%, these two cases resulted in higher F1 scores. Moreover, the S3D model, which is our proposed method using both keyframes and an SE block, showed the best performance, with an F1 score of 97.14%.

Table 5 presents the experimental results on the effective improvement of the model performance depending on the number of SE blocks and locations of the SE blocks in the S3D model. Based on the experimental results, the F1 score obtained when the SE block was added before or after all pooling layers with five SE blocks ranged between 95.65% and 95.77%, and compared with the F1 score of 95.77%, obtained without the addition of the SE block, the model performance was not improved. Furthermore, based on the results of comparing the performance with different locations of the SE block in each pooling layer, the F1 score of our proposed method, in which the SE block was added immediately after the 5th pooling layer, was 97.14%, indicating the best performance.

Table 4. Experimental results of the effect of keyframes and the SE block in the TSF dataset (unit: %).

Method	Accuracy	Recall	Precision	F1 Score
C3D	90.91	94.29	94.29	94.29
keyframes + C3D	93.18	97.14	94.44	95.77
C3D + SE block	93.18	94.29	97.06	95.65
keyframes + C3D + SE block (Proposed method)	95.45	97.14	97.14	97.14

Table 5. Experimental results of the effect of the SE block location on the model performance in the TSF dataset (unit: %).

Number of SE Blocks	Location of the SE Block	Accuracy	Recall	Precision	F1 Score
0	Without SE block	93.18	97.14	94.44	95.77
5	Before every pooling layer	93.18	94.29	97.06	95.65
	After every pooling layer	93.18	97.14	94.44	95.77
1	Before the 1st pooling layer	93.18	97.14	94.44	95.77
	After the 1st pooling layer	93.18	94.29	97.06	95.65
	Before the 2nd pooling layer	93.18	94.29	97.06	95.65
	After the 2nd pooling layer	93.18	97.14	94.44	95.77
	Before the 3rd pooling layer	90.91	94.29	94.74	94.51
	After the 3rd pooling layer	93.18	94.29	97.06	95.65
	Before the 4th pooling layer	93.18	97.14	94.44	95.77
	After the 4th pooling layer	93.18	97.14	94.44	95.77
	Before the 5th pooling layer	93.18	94.29	97.06	95.65
	After the 5th pooling layer (proposed method)	95.45	97.14	97.14	97.14

Table 6 presents the performance of the S3D model, depending on the number of key frames. Based on the experimental results, the F1 score with 8 key frames was 94.44%, that with 16 key frames was 97.14%, and that with 32 key frames was 97.06%, indicating that the performance with 16 key frames is the best.

Table 6. Comparison of performance of the number of key frames in the TSF dataset (unit: %).

Number of Key Frames	Accuracy	Recall	Precision	F1 Score
8	93.18	97.14	91.89	94.44
16 (proposed method)	95.45	97.14	97.14	97.14
32	95.45	94.29	100.00	97.06

Comparisons with State-of-the-Art Methods

In this study, for the comparison with state-of-the-art methods, experiments were carried out using handcrafted-based existing studies [21] and deep learning-based existing studies [5,6,24]. Because there are not many cases of investigating FDSs using thermal videos, the deep learning-based methods resulting in a good performance in the action recognition field [5,6] were also utilized for the comparative experiments. As presented in Table 7, it was identified that our proposed method shows a higher recognition performance compared with existing state-of-the-art methods in the TSF dataset.

Table 7. Comparison with state-of-the-art methods on the TSF dataset (unit: %).

	Method	Accuracy	Recall	Precision	F1 Score
Handcrafted-based	Optical flow + FFT + SVM [21]	72.73	88.57	79.49	83.78
Deep learning-based	SlowFast Networks [6]	88.64	91.43	94.12	92.76
	C3D [5]	90.91	94.29	94.29	94.29
	Keyframes extraction + 3D CNN [24]	79.55	91.43	84.21	87.67
	Proposed method	95.45	97.14	97.14	97.14

4.3.2. TCL Dataset

Ablation Studies

Table 8 shows the ablation study results on using the keyframes as the network input in the C3D, a 3D CNN model that is the foundation of the S3D model proposed in this study, and adding the SE block to the network. As shown in Table 8, the F1 score of the case in which the keyframes were used as the network input in the C3D model was found to be 88.68%, and that of the case where the SE block was added to the C3D model was 80.99%. Compared with the F1 score of the case using only the C3D model, which was 79.84%, these two cases resulted in higher F1 scores. Furthermore, the S3D model, which is our proposed method using both keyframes and an SE block, showed the best performance, with an F1 score of 95.30%.

Table 8. Experimental results of the effect of keyframes and SE block in the TCL dataset (unit: %).

Method	Accuracy	Recall	Precision	F1 Score
C3D	84.35	92.38	70.29	79.84
keyframes + C3D	92.16	91.43	86.10	88.68
C3D + SE block	85.30	93.33	71.53	80.99
keyframes + C3D + SE block (Proposed method)	96.89	94.06	96.59	95.30

Table 9 presents the experimental results on the effective improvement of the model performance depending on different numbers of SE blocks and locations of the SE block in the S3D model. Based on the experimental results, the F1 score obtained after the addition of the SE block showed a higher F1 score at all times compared with the case with no addition of SE block, regardless of the location of the SE block. Moreover, when the SE block was added to the rear pooling layer rather than the front pooling layer, the performance was more significantly improved. This can be attributed to the fact that the feature map obtained from the rear pooling layer has better information to classify the classes channel-wise. Furthermore, the range of the F1 score obtained after adding the SE block was 92.95–95.30%, suggesting significantly improved performance compared to the F1 score of 88.68% obtained with no addition of an SE block. The F1 score of our proposed method, in which the SE block was added immediately after the last 5th pooling layer, was found to be 95.30%, indicating the best performance.

Table 10 presents the performance of the S3D model depending on the number of key frames. Based on the experimental results, the F1 score with 8 key frames was 92.49%, that with 16 key frames was 95.30%, and that with 32 key frames was 94.14%, indicating that the performance with 16 key frames is the best.

Table 9. Experimental results of the effect of the SE block location on the model performance in the TCL dataset (unit: %).

Number of SE Blocks	Location of the SE Block	Accuracy	Recall	Precision	F1 Score
0	Without SE block	92.16	91.43	86.10	88.68
5	Before every pooling layer	95.37	90.51	95.52	92.95
	After every pooling layer	95.61	91.92	94.88	93.38
1	Before the 1st pooling layer	95.45	93.58	92.98	93.28
	After the 1st pooling layer	95.69	91.69	95.32	93.47
	Before the 2nd pooling layer	95.29	90.25	97.40	93.69
	After the 2nd pooling layer	95.45	91.92	94.44	93.16
	Before the 3rd pooling layer	95.93	90.25	97.40	93.69
	After the 3rd pooling layer	95.85	92.16	95.35	93.73
	Before the 4th pooling layer	96.25	92.16	96.56	94.31
	After the 4th pooling layer	96.25	93.58	95.20	94.38
	Before the 5th pooling layer	96.57	93.82	95.88	94.84
	After the 5th pooling layer (proposed method)	96.89	94.06	96.59	95.30

Table 10. Comparison of performance of the number of key frames in the TCL dataset (unit: %).

Number of Key Frames	Accuracy	Recall	Precision	F1 Score
8	95.01	93.33	91.67	92.49
16 (proposed method)	96.89	94.06	96.59	95.30
32	96.17	96.50	91.90	94.14

Comparisons with State-of-the-Art Methods

Table 11 presents the comparison of the performance of the S3D model proposed in this study with that of state-of-the-art methods. As presented in Table 11, it was identified that our proposed method showed a higher recognition performance compared with existing state-of-the-art methods in the TCL dataset. In particular, our proposed method showed 6.65% higher accuracy, 4.06% higher recall, 14.06% higher precision, and a 9.2% higher F1 score compared to the existing state-of-the-art methods.

Table 11. Comparison with the state-of-the-art methods on the TCL dataset (unit: %).

	Method	Accuracy	Recall	Precision	F1 Score
Handcrafted-based	Optical flow + FFT + SVM [21]	74.44	72.86	59.77	65.67
Deep learning-based	SlowFast Networks [6]	77.64	86.19	61.99	72.11
	C3D [5]	84.35	92.38	70.29	79.84
	Keyframes extraction + 3D CNN [24]	90.24	90.00	82.53	86.10
	Proposed method	96.89	94.06	96.59	95.30

4.3.3. eHomeSenior Dataset

Ablation Studies

Table 12 presents the ablation study results on using the keyframes as the network input in the C3D, a 3D CNN model that is the foundation of the S3D model proposed in

this study, and adding the SE block to the network. As shown in Table 12, the F1 score of the case in which the keyframes were utilized as the network input in the C3D model was 96.74%, and that of the case in which the SE block was added to the C3D model was 96%. Compared with the F1 score of the case using only the C3D model, which was found to be 95.28%, these two cases resulted in higher F1 scores. Moreover, the S3D model, which is our proposed method using both keyframes and an SE block, showed the best performance with an F1 score of 98.89%.

Table 12. Experimental results of the effect of keyframes and SE block in the eHomeSenior dataset (unit: %).

Method	Accuracy	Recall	Precision	F1 Score
C3D	95.28	97.76	92.93	95.28
keyframes + C3D	96.81	96.36	97.13	96.74
C3D + SE block	95.79	97.52	94.53	96.00
keyframes + C3D + SE block (Proposed method)	98.91	98.46	99.33	98.89

Table 13 shows the experimental results on the effective improvement of the model performance depending on different numbers of SE blocks and locations of the SE block in the S3D model. Based on the experimental results, the F1 score obtained after the addition of the SE block showed a higher F1 score at all times compared with the case with no addition of an SE block, regardless of the location of the SE block. Furthermore, the F1 score of our proposed method, in which the SE block was added immediately after the 5th pooling layer, was found to be 98.89%, indicating the best performance.

Table 13. Experimental results of the effect of the SE block location on the model performance in the eHomeSenior dataset (unit: %).

Number of SE Blocks	Location of the SE Block	Accuracy	Recall	Precision	F1 Score
0	Without SE block	96.81	96.36	97.13	96.74
5	Before every pooling layer	97.31	98.38	96.18	97.27
	After every pooling layer	97.59	97.56	97.46	97.51
1	Before the 1st pooling layer	97.45	97.32	94.63	95.96
	After the 1st pooling layer	97.59	97.45	97.66	97.55
	Before the 2nd pooling layer	97.79	97.30	98.08	97.69
	After the 2nd pooling layer	97.45	97.51	97.42	97.46
	Before the 3rd pooling layer	97.53	96.84	98.08	97.46
	After the 3rd pooling layer	97.45	98.35	96.49	97.41
	Before the 4th pooling layer	96.85	96.90	96.71	96.80
	After the 4th pooling layer	97.65	97.48	97.66	97.57
	Before the 5th pooling layer	97.49	97.79	97.16	97.47
	After the 5th pooling layer (proposed method)	98.91	98.46	99.33	98.89

Table 14 presents the performance of the S3D model depending on the number of key frames. Based on the experimental results, the F1 score with 8 key frames was 97.81%, that with 16 key frames was 98.89%, and that with 32 key frames was 98.02%, indicating that the performance with 16 key frames is the best.

Table 14. Comparison of performance of the number of key frames in the eHomeSenior dataset (unit: %).

Number of Key Frames	Accuracy	Recall	Precision	F1 Score
8	97.83	96.54	99.11	97.81
16 (proposed method)	98.91	98.46	99.33	98.89
32	98.05	96.96	99.11	98.02

Comparisons with State-of-the-Art Methods

Table 15 shows the comparison of the performance of the S3D model proposed in this study with that of state-of-the-art methods. As shown in Table 15, it was identified that our proposed method showed higher recognition performance compared with existing state-of-the-art methods in the eHomeSenior dataset. In particular, our proposed method showed 2.99% higher accuracy, 2.41% higher recall, 3.31% higher precision, and a 2.86% higher F1 score compared to the existing state-of-the-art methods.

Table 15. Comparison with the state-of-the-art methods on the eHomeSenior dataset (unit: %).

	Method	Accuracy	Recall	Precision	F1 Score
Handcrafted-based	Optical flow + FFT + SVM [21]	76.37	75.33	75.67	75.50
Deep learning-based	SlowFast Networks [6]	94.20	95.29	93.98	94.63
	C3D [5]	95.28	97.76	92.93	95.28
	Keyframes extraction + 3D CNN [24]	95.92	96.05	96.02	96.03
	Proposed method	98.91	98.46	99.33	98.89

4.3.4. Comparative Processing Complexities of the Proposed Method and the State-of-the-Art Methods

In the next experiments, the comparative processing time of the proposed method and the state-of-the-art methods were measured in the environment with a desktop computer including an NVIDIA GeForce GTX Titan X GPU and the Jetson TX2 board (NVIDIA Corp., Santa Clara, CA, USA) [48] (Figure 7), as described in Section 4.1, and the results are shown in Table 16. The NVIDIA GeForce GTX Titan X GPU has 3072 CUDA cores and 12 GB memory, and the Jetson TX2 board has 256 CUDA cores, 8 GB 128-bit LPDDR4 memory, and a dual-core NVIDIA Denver 2 64-bit CPU. The power consumption of the Jetson TX2 board is less than 7.5 W. The proposed method was ported with Pytorch 1.7.1 [39] in Ubuntu 18.04 OS. The versions of the installed framework and library include Python 3.8.5; NVIDIA CUDA® toolkit [40] and NVIDIA CUDNN [41] versions are 10.2 and 8.0, respectively.

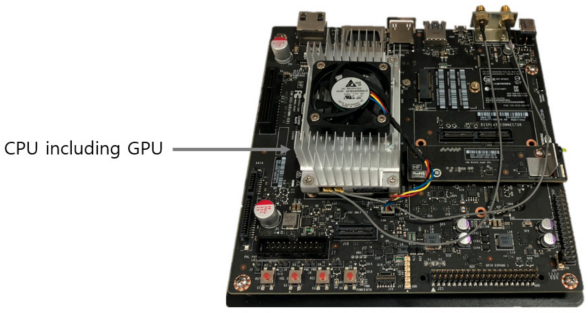


Figure 7. Jetson TX2 board.

The processing time per frame of the desktop computer for the TSF dataset and TCL dataset that are inputs to the S3D model with the resolution of 112 × 112 pixels

was 45.37 ms, and the processing time per frame of the Jetson TX2 board was 168.62 ms. Moreover, the processing time per frame of the desktop computer for the eHomeSenior dataset that is input to the S3D model with the resolution of 32 × 24 pixels was 6.68 ms, and the processing time per frame of the Jetson TX2 board was 23.18 ms. Based on these, it was identified that the method proposed in this study shows the processing speed of 22.04 (1000/45.37) frames per second (fps) to 149.7 (1000/6.68) fps on the desktop computer. Furthermore, it was identified as having a processing speed of 5.93 (1000/168.62) fps to 43.14 (1000/23.18) fps in the Jetson TX2. The training of our algorithm was performed on the desktop computer. Then, the trained algorithm was transferred to the Jetson TX2 embedded system, and it could only be operated on the embedded system without training. Therefore, the processing speed was as fast as 5.93~43.14 fps on the embedded system, as shown in Table 16. Based on these results, it was verified that the method proposed in this study is applicable to embedded systems with limited computing resources and power. In addition, the comparative processing time of proposed and the state-of-the-art methods are shown in Table 16. Although the processing speed of proposed method was slightly slower than other methods, our method shows a higher accuracy of fall detection than the state-of-the-art methods, as shown in Tables 7, 11 and 15.

Table 16. Comparative processing time of the proposed method and the state-of-the-art methods.

Method	Environment	Processing Time on TSF, TCL Dataset	Processing Time on eHomeSenior Dataset
Optical flow + FFT + SVM [21]	Desktop	20.35 ms (49.14 fps)	9.4 ms (106.38 fps)
	Jetson TX2	104.21 ms (9.60 fps)	41.63 ms (24.02 fps)
SlowFast Networks [6]	Desktop	35.82 ms (27.92 fps)	5.6 ms (178.57 fps)
	Jetson TX2	136.47 ms (7.33 fps)	20.89 ms (47.87 fps)
C3D [5]	Desktop	45.09 ms (22.18 fps)	6.64 ms (150.6 fps)
	Jetson TX2	167.11 ms (5.98 fps)	22.57 ms (44.31 fps)
Keyframes extraction + 3D CNN [24]	Desktop	30.02 ms (33.31 fps)	4.39 ms (227.79 fps)
	Jetson TX2	117.38 ms (8.52 fps)	17.03 ms (58.72 fps)
Proposed method	Desktop	45.37 ms (22.04 fps)	6.68 ms (149.7 fps)
	Jetson TX2	168.62 ms (5.93 fps)	23.18 ms (43.14 fps)

Table 17 presents the comparative Giga floating operations per second (GFLOPS) required for operating our proposed method and the state-of-the-art methods in each dataset. With the TSF and TCL datasets, which include high-resolution images, the GFLOPS of our proposed method was 75.82, and the GFLOPS of our proposed method with the eHomeSenior dataset, which includes low-resolution images, was 4.62, which confirms that the computation is significantly reduced in low-resolution images. Although the GFLOPS of the proposed method was slightly larger than other methods, our method shows a higher accuracy of fall detection than the state-of-the-art methods, as shown in Tables 7, 11 and 15.

Table 17. Comparative GFLOPS of the proposed method and the state-of-the-art methods.

Method	With TSF and TCL Datasets	With eHomeSenior Dataset
Optical flow + FFT + SVM [21]	34.91	6.61
SlowFast Networks [6]	62.17	3.97
C3D [5]	75.35	4.61
Keyframes extraction + 3D CNN [24]	51.22	3.15
Proposed method	75.82	4.62

4.4. Discussion

Figure 8 shows the case with the correct detection of a fall and ADL cases using the method proposed in this study. As shown in this figure, it was identified that even in the case of a fall that occurs while getting out of bed and ADL that sits down, the correctly detected results were obtained by the proposed method.

Figure 9 presents the case with the incorrect detection of a fall and ADL cases using the method proposed in this study. Figure 9a shows the case of incorrectly recognizing a fall case as an ADL case, and Figure 9b shows the case of incorrectly recognizing an ADL case as a fall case. As presented in this figure, the case of incorrectly recognizing a fall case as an ADL case took place when a person fell in the opposite direction to the camera, and that of incorrectly recognizing an ADL case as a fall case occurred when a person changed his posture on the bed.

Unlike a fall, which can be defined by a few cases, such as standing and falling or sitting and falling, ADL indicates all situations except for a fall; hence, the pattern of fall videos is relatively simple, and the data diversity is low. Therefore, it can be expected that if the learned model distinguishes these characteristics of a fall video, the prediction performance will be improved. Figure 10 shows graphs of activation values by channel of the SE block in fall and ADL videos obtained during the learning of the S3D model. As shown in these graphs, the activation by channel of the fall video shows a small variation, whereas the activation by channel of the ADL video shows a relatively larger variation. Such a difference suggests that the diversity of the fall video data is relatively smaller than that of the ADL video, and the SE block identified these characteristics of the fall video well. It can be interpreted that the SE block differently applies the location and intensity of the channel emphasized in fall and ADL videos and provides important information for fall detection.

Because it is difficult to rely on and use the learned model without knowing on which part the model draws a conclusion, in this study, a basis for fall detection of the S3D model was analyzed by visualizing the activation maps based on the features learned using Grad-CAM [49]. Figure 11 presents the results of the activation maps visualized using the keyframes extracted from the image sequence samples of the TSF, TCL, and eHomeSenior datasets alongside Grad-CAM. In the activation maps, the areas with a high feature value have a bright pixel value close to red, and the areas with a low feature value have a dark pixel value close to blue.

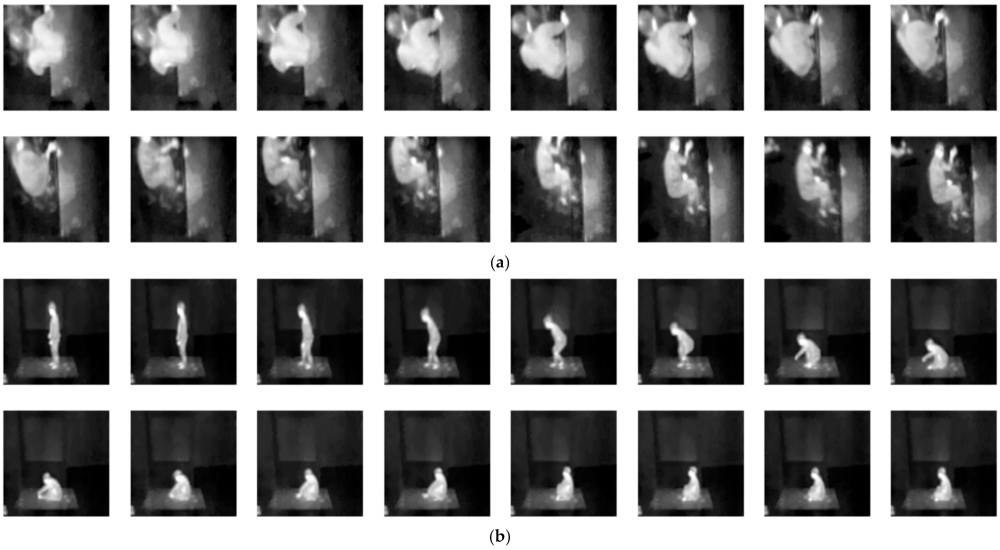


Figure 8. Correct detection of (**a**) fall and (**b**) ADL cases.

Based on the keyframes and activation maps presented in Figure 11, it was found that the learned S3D model can accurately identify the area where a person is located and their

movement. Because it is a key to accurately identify the information on the location and movement of a person for fall detection, it was concluded on the basis of these results that the S3D model was well-learned.

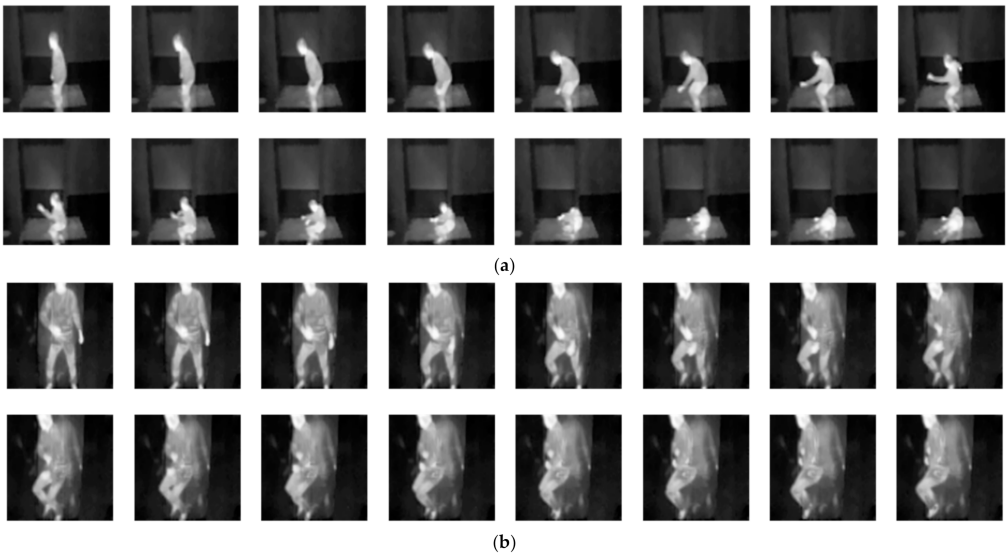


Figure 9. Incorrect detection of (**a**) fall and (**b**) ADL cases.

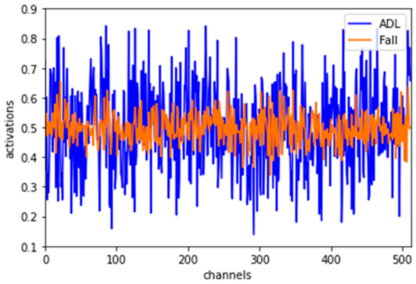


Figure 10. Activation intensity by the channel of the SE block in fall and ADL videos.

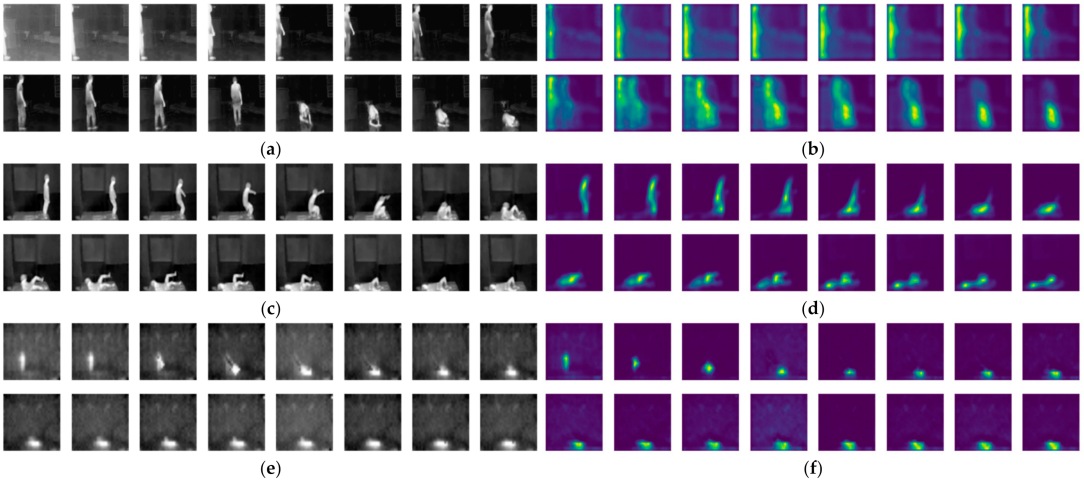


Figure 11. Example of activation maps. (**a**) Keyframes of the TSF dataset; (**b**) activation map results of (**a**); (**c**) keyframes of the TCL dataset; (**d**) activation map results of (**c**); (**e**) keyframes of the eHomeSenior dataset; (**f**) activation map results of (**e**).

As a last discussion, the theoretical comparison between the existing studies on FDSs (explained in Section 2) and the proposed method are tabulated in Table 18.

Table 18. Comparisons of previous and proposed methods on fall detection.

Categories		Method	Advantages	Disadvantages
Wearable device-based	Accelerometer	Magnitude thresholding [8]	It has a very low calculation volume, and hence, it can use a low-specification processor.	As it only considers the magnitude, there is a high probability of faulty detection in the presence of a dynamic movement.
		Decision tree + state machine [9]	As it learns the thresholding value from the data, the data characteristics are well-reflected.	It still depends on the learned fixed threshold value, and the sensor should be attached to the body or be worn.
	Fusing accelerometer and gyroscope	Thresholding of the changes of acceleration and angle [10,11]	It is relatively more accurate than using a single sensor.	- It could not detect slow falling. - The sensor should be attached to the body or be worn.
Ambient device-based	Radar	Autoencoder + Logistic Regression [14]	It detects well at both long and short distances, and it has fewer concerns about privacy invasion compared with cameras.	The sensor is relatively expensive.
	Ultrasonic	Event pattern matching [15]	The sensor price is low, and there are fewer concerns about privacy invasion.	There is a high probability of misclassification in cases in which there exist other moving objects, such as pets or robotic vacuums.
Vision-based	Visible light camera	Foreground extraction + optical flow + KNN [17]	The spatial information can be utilized for the prediction.	It is vulnerable to privacy problems.
	Depth camera	ROI detection + thresholding of the changes of pixel positions [18]	As it is not a training-based method, it does not require much data for building a model.	Because it does not consider the moving speed or direction, errors could occur in the actual environment.
		Outline detection + thresholding of tangential distribution [19]	The spatial information can be well-detected, even in the dark.	If there are many other objects nearby, the accuracy could decrease.
	Thermal camera	Histogram + SVM [20]	Because of the fewer features used, the calculation speed is relatively fast.	The accuracy cannot be guaranteed if the environment is different from the environment where the training data were obtained.
		Optical flow + FFT + SVM [21]	The accuracy change is not significant in the databases in different environments.	The accuracy of the fall detection is low.
		ConvLSTM + autoencoder [22,23]	It is a network with a recurrent structure and can extract the temporal information well.	If there is a movement pattern that was not learned with the training dataset, the risk of misclassification increases.
		Keyframes extraction + 3D CNN [24]	It is efficient because it uses only keyframes as the network input.	- The calculation of optical flow for extracting keyframes is slow. - The accuracy decreases under various environmental changes.
		Keyframes extraction + 3D CNN with SE block (Proposed)	The fall detection is conducted at a high accuracy by considering the spatiotemporal information through the SE block.	The calculation of optical flow for extracting keyframes is slow.

5. Conclusions

In this study, we newly proposed an S3D-based FDS method. In the proposed method, keyframes are extracted using the magnitudes of the optical flow vectors in input thermal videos, the extracted keyframes are used as the input of the 3D CNN with an SE block applied, and fall detection is carried out based on the output values of the 3D CNN.

In this study, comparative experiments were performed using three open databases of thermal videos with different image resolutions. Our proposed method obtained F1 scores of 97.14%, 95.30%, and 98.89% in the TSF, TCL, and eHomeSeniors datasets, respectively, and it was identified that these results are superior to those obtained using state-of-the-art methods. Moreover, it was confirmed that the proposed method proposed is applicable not only to desktop computers, but also to embedded systems with limited computing power and resources. On the basis of an activation map analysis through Grad-CAM, it was identified that the proposed S3D model was learned such that it could extract the features for fall detection well.

The reason for the higher recognition accuracy in the eHomeSeniors dataset with low resolution compared with the other two databases is that there were relatively fewer background noise components apart from the target in the input video, owing to the low video resolution. However, the method proposed in this study resulted in an error in the case of falling in the opposite direction to the camera.

In future work, we would research fall detection methods robust to the various directions of the fall based on the camera. In addition, we would research fall detection methods in severe environments of extremely low resolution or optical and motion blurring in the captured thermal images.

Author Contributions: Methodology, S.B.H.; conceptualization, Y.H.K.; validation, S.H.N.; supervision, K.R.P.; writing—original draft, S.B.H.; writing—editing and review, K.R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) through the Basic Science Research Program (NRF-2021R1F1A1045587), in part by the NRF funded by the MSIT through the Basic Science Re-search Program (NRF-2020R1A2C1006179), and in part by the MSIT, Korea, under the ITRC (In-formation Technology Research Center) support program (IITP-2021-2020-0-01789) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. World Health Organization. WHO Global Report on Falls Prevention in Older Age. Available online: https://www.who.int/ageing/publications/Falls_prevention7March.pdf (accessed on 26 August 2021).
2. Fleming, J.; Brayne, C. Inability to get up after falling, subsequent time on floor, and summoning help: Prospective cohort study in people over 90. *BMJ* **2008**, *337*, a2227. [CrossRef] [PubMed]
3. Wang, Z.; Ramamoorthy, V.; Gal, U.; Guez, A. Possible life saver: A review on human fall detection technology. *Robotics* **2020**, *9*, 55. [CrossRef]
4. Hayashida, A.; Moshnyaga, V.; Hashimoto, K. The use of thermal IR array sensor for indoor fall detection. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Banff, AB, Canada, 5–8 October 2017; pp. 594–599.
5. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
6. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast networks for video recognition. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6202–6211.
7. S3D. Available online: https://github.com/baek2sm/S3D (accessed on 26 August 2021).

8. Kostopoulos, P.; Nunes, T.; Salvi, K.; Deriaz, M.; Torrent, J. Increased fall detection accuracy in an accelerometer-based algorithm considering residual movement. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, Lisbon, Portugal, 10–13 January 2015; Volume 2, pp. 30–36.
9. Aguiar, B.; Rocha, T.; Silva, J.; Sousa, I. Accelerometer-based fall detection for smartphones. In Proceedings of the IEEE International Symposium on Medical Measurements and Applications, Lisbon, Portugal, 11–12 June 2014; pp. 1–6.
10. Rakhman, A.Z.; Nugroho, L.E. Fall detection system using accelerometer and gyroscope based on smartphone. In Proceedings of the 1st International Conference on Information Technology, Computer, and Electrical Engineering, Semarang, Indonesia, 7–8 November 2014; pp. 99–104.
11. Torres, G.G.; Henriques, R.V.B.; Pereira, C.E.; Müller, I. An EnOcean wearable device with fall detection algorithm integrated with a smart home system. *Int. Fed. Autom. Control* **2018**, *51*, 9–14. [CrossRef]
12. Pourbemany, J.; Zhu, Y.; Bettati, R. Breath to Pair (B2P): Respiration-based pairing protocol for wearable devices. *arXiv* **2021**, arXiv:2107.11677.
13. Pourbemany, J.; Zhu, Y.; Bettati, R. A survey of wearable devices pairing based on biometric signals. *arXiv* **2021**, arXiv:2107.11685.
14. Jokanović, B.; Amin, M. Fall detection using deep learning in range-doppler radars. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 180–189. [CrossRef]
15. Chang, Y.-T.; Shih, T.K. Human fall detection based on event pattern matching with ultrasonic array sensors. In Proceedings of the 10th International Conference on Ubi-Media Computing and Workshops, Pattaya, Thailand, 1–4 August 2017; pp. 1–4.
16. Sanaat, A.; Zaidi, H. Depth of interaction estimation in a preclinical PET scanner equipped with monolithic crystals coupled to SiPMs using a deep neural network. *Appl. Sci.* **2020**, *10*, 4753. [CrossRef]
17. De Miguel, K.; Brunete, A.; Hernando, M.; Gambao, E. Home camera-based fall detection system for the elderly. *Sensors* **2017**, *17*, 2864. [CrossRef] [PubMed]
18. Sase, P.S.; Bhandari, S.H. Human fall detection using depth videos. In Proceedings of the 5th International Conference on Signal Processing and Integrated Networks, Noida, India, 22–23 February 2018; pp. 546–549.
19. Kong, X.; Meng, L.; Tomiyama, H. Fall detection for elderly persons using a depth camera. In Proceedings of the International Conference on Advanced Mechatronic Systems, Xiamen, China, 6–9 December 2017; pp. 269–273.
20. Song, K.-S.; Nho, Y.-H.; Kwon, D.-S. Histogram based fall prediction of patients using a thermal imagery camera. In Proceedings of the 14th International Conference on Ubiquitous Robots and Ambient Intelligence, Jeju, Korea, 28 June–1 July 2017; pp. 161–164.
21. Vadivelu, S.; Ganesan, S.; Murthy, O.R.; Dhall, A. Thermal imaging based elderly fall detection. In Proceedings of the Asian Conference on Computer Vision International Workshops, Taipei, Taiwan, 20–24 November 2016; pp. 541–553.
22. Nogas, J.; Khan, S.S.; Mihailidis, A. Fall detection from thermal camera using convolutional lstm autoencoder. In Proceedings of the International Joint Conference on Artificial Intelligence Workshop, Stockholm, Sweden, 13–19 July 2018.
23. Elshwemy, F.A.; Elbasiony, R.; Saidahmed, M.T. A new approach for thermal vision based fall detection using residual autoencoder. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 250–258. [CrossRef]
24. Kim, D.-E.; Jeon, B.; Kwon, D.-S. 3D convolutional neural networks based fall detection with thermal camera. *J. Korea Robot. Soc.* **2018**, *13*, 45–54. [CrossRef]
25. Bangyal, W.H.; Ahmad, J.; Rauf, H.T. Optimization of neural network using improved bat algorithm for data classification. *J. Med. Imaging Health Inform.* **2019**, *9*, 670–681. [CrossRef]
26. Rauf, H.T.; Bangyal, W.H.; Ahmad, J. Training of artificial neural network using PSO with novel initialization technique. In Proceedings of the International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, Sakhier, Bahrain, 18–20 November 2018; pp. 1–8.
27. Castellano, G.; Castiello, C.; Mencar, C.; Vessio, G. Crowd detection in aerial images using spatial graphs and fully-convolutional neural networks. *IEEE Access* **2020**, *8*, 64534–64544. [CrossRef]
28. Yao, L.; Xiao, X.; Cao, R.; Chen, F.; Chen, T. Three stream 3D CNN with SE block for micro-expression recognition. In Proceedings of the IEEE International Conference on Computer Engineering and Application, Guangzhou, China, 18–20 March 2020; pp. 439–443.
29. Han, Y.; Wei, C.; Zhou, R.; Hong, Z.; Zhang, Y.; Yang, S. Combining 3D-CNN and squeeze-and-excitation networks for remote sensing sea ice image classification. *Math. Probl. Eng.* **2020**, *2020*, 8065396. [CrossRef]
30. Lee, W.O.; Lee, E.C.; Park, K.R. Blink detection robust to various facial poses. *J. Neurosci. Methods* **2010**, *193*, 356–372. [CrossRef] [PubMed]
31. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
32. Baker, S.; Matthews, I. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255. [CrossRef]
33. Bansal, M.; Kumar, M.; Kumar, M.; Kumar, K. An efficient technique for object recognition using Shi-Tomasi corner detection algorithm. *Soft Comput.* **2021**, *25*, 4423–4432. [CrossRef]
34. Beauchemin, S.; Barron, J. The computation of optical flow. *ACM Comput. Surv.* **1995**, *27*, 433–466. [CrossRef]
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
36. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.

37. Riquelme, F.; Espinoza, C.; Rodenas, T.; Minonzio, J.-G.; Taramasco, C. eHomeSeniors dataset: An infrared thermal sensor dataset for automatic fall detection research. *Sensors* **2019**, *19*, 4565. [CrossRef] [PubMed]
38. OpenCV. Available online: https://opencv.org (accessed on 10 September 2021).
39. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 8026–8037.
40. NVIDIA CUDA Toolkit. Available online: https://developer.nvidia.com/cuda-toolkit (accessed on 10 September 2021).
41. NVIDIA cuDNN. Available online: https://developer.nvidia.com/cudnn (accessed on 10 September 2021).
42. NVIDIA Geforce TITAN X Graphics Card. Available online: https://www.nvidia.com/en-us/geforce/graphics-cards/geforce-gtx-titan-x (accessed on 10 September 2021).
43. Keren, G.; Sabato, S.; Schuller, B. Fast single-class classification and the principle of logit separation. In Proceedings of the IEEE International Conference on Data Mining, Singapore, 17–20 November 2018; pp. 227–236.
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2017**, arXiv:1412.6980.
46. Confusion Matrix. Available online: https://en.wikipedia.org/wiki/Confusion_matrix (accessed on 14 January 2022).
47. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
48. Jetson TX2 Board. Available online: https://developer.nvidia.com/embedded/jetson-tx2 (accessed on 26 August 2021).
49. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.