AJEMA GLOBAL RESEARCH NETWORK

AMERICAN Journal of Engineering, Mechanics and Architecture
Volume 3, Issue 10, 2025 ISSN (E): 2993-2637

Fall-Detection Model for Elderly People to Improve Safety Based on Enhancing Inception v3 and LSTM

Qasim Mahdi Haref[1,2], Murteza Hanoon Tuama[2*], Wahhab Muslim Mashloosh[2]
[1]*School of Computer Science and Engineering, Central South University, Changsha 410083, China*
[2]*Department of Computer Techniques Engineering, Imam Al-Kadhum University College, Baghdad, Iraq*
[*]*Correspondence: murtezahanoon@iku.edu.iq*

Abstract: Fall detection, especially among the elderly, is crucial due to the potentially severe consequences of the delayed identification of falls. This paper introduces an innovative approach to enhance the Inception v3 model. It involves adding the feature extractor ends with a global average pooling layer to address overfitting issues and eliminate the SoftMax layer. Additionally, LSTM is incorporated to improve classification accuracy by preprocessing the data and removing the background, thereby reducing confusion and boosting accuracy. The initial steps include obtaining a dataset from video footage, converting it to a series of images, and subjecting it to preprocessing for the next stages of training and testing. Our model was trained and tested using three distinct datasets: our dataset, the fall-detection dataset, and the Le2i dataset. The extracted features were fed into the Long Short-Term Memory (LSTM) classifier for fall detection classification. The LSTM classifier leverages these features to distinguish between fall and non-fall instances. The proposed model achieves significant accuracy, with scores of 0.98 on our dataset, 0.97 on the Le2i dataset, and 0.96 on the Fall Detection Dataset, underscoring its effectiveness in robust fall detection. This study contributes to fall detection, particularly in scenarios crucial to the well-being of the elderly population.

Keywords: Fall Detection, Inception V3, LSTM, Classification, Computer Vision, Image Preprocessing

Introduction

Home accidents can occur across all age groups but are particularly common among the elderly and those in poor health. Accidents can result in severe injury or death. The expenses associated with medical care or hospitalisation increase one's financial and time burden. According to a study by Bahar et al.[1], the average age of the individuals who experienced a fall was 80.5 ±8.3. Of the total number of patients, 80 individuals, accounting for 60.6% of the sample, were younger than 85.

On the other hand, 52 patients, making up 39.4% of the sample, were aged 85 years and older. Of the total number of patients, 62 (47%) were identified as fragile. Sixty-seven individuals, accounting for 50.8% of the total, feared falling, and 62 (47%) exhibited polypharmacy. The oldest elderly group had a considerably greater frequency of diabetes mellitus, frailty, fear of falling, walking-aid use, and the need for a regular carer. Therefore, it is evident that the decline of the aged results solely in adverse outcomes. Continuous care is crucial for this population.

The World Health Organization (WHO) conducted a survey study in 2011 that found 650 million working-age people with disabilities [2]. Older people and people with disabilities must be monitored twenty-four hours a day to provide rapid assistance in the event of an accident. In the

United States, 20% to 30% of the elderly suffer from falls and are exposed to bruises and injuries, so it is critical to provide them with first aid as soon as possible. For this purpose, the fall-detection system has become essential in nursing homes.

The increasing prevalence of surveillance cameras and the widespread dissemination of video clips on the internet have posed challenges regarding human monitoring capabilities. Recently, there has been a significant use of deep learning techniques to discern human movements captured by surveillance cameras. Due to this rationale, many researchers work in this domain [3-5]. Some of these studies focus on specific measures using different models and sensors: falling [6], sitting [7], standing [8], running [9], and walking [10]. Fall detection has become one of the most important areas of artificial intelligence that specializes in human movement recognition, especially for the elderly, because it is difficult to monitor them until they receive first aid before the injury worsens, especially if the injury from the fall is severe.

Falls can occur due to several circumstances, such as muscular weakness, insufficient blood circulation, or dizziness, and can lead to severe injuries, such as brain hemorrhage, particularly among older individuals. Timely medical intervention is essential for reducing the severity of injuries caused by falls and improving the likelihood of survival. Abnormal-action detection is a vast area of research because it is related to security in the city when used in surveillance camera systems and monitoring human actions. We used the same anomaly-detection algorithm but focus exclusively on falling and other actions relevant in this area. We left the other actions to increase accuracy in our work, and many actions did not need it. Fall detection is an abnormal human action; therefore, all anomaly detection considers falling an abnormal action but does not focus on it. Chongke et al. [11] created a model using high-level characteristics for computer vision models for object detection and classification, and other researchers have studied abnormal action and the detection of falling with many actions.A falling accident can cause incapacity or cripple, and the danger increases when the person is alone and cannot call or inform others. A falling accident may also cause loss of consciousness; therefore, monitoring older adults is important so that assistance and first aid can be provided. In recent years, many studies have focused on helping make the elderly safer using different technologies to monitor them 24 hours a day [12-14]. This technology allows people to live safely and freely in their homes, and it is used in most developed countries.The Inception v3 structure breaks down huge kernels of convolution into smaller ones, resulting in superior parallel convolution capability and a robust model expression capacity. It can handle high-performance computing tasks using matrices with high density and can handle a larger quantity of intricate and varied spatial data. Applying the Inception v3 model to identify the action of human types will result in overfitting, less training efficiency, and other issues. Therefore, more improvements to the model are necessary.

In recent years, artificial intelligence has become widely used to distinguish human movement and has created several algorithms, but few have focused on monitoring the elderly. A human action recognition algorithm with high precision, accuracy, and activity-classification recognisability is required to monitor older adults. Artificial-intelligence algorithms can be used to predict and classify the next move. Moreover, the main contributions of this paper are:

a. It constructs a new model for fall detection in elderly people using Inception v3 and LSTM.

b. The proposed model is an enhanced version of the Inception v3 paradigm. It improves upon the original model by using smaller convolution kernels instead of larger ones, which allows for greater expressive power. Additionally, it supports parallel convolution and incorporates a dropout layer to effectively prevent overfitting.

c. It preprocesses data to remove superfluous elements and attributes from the image, retaining only the essential features required to ascertain the individual's posture.

d. It finds a new dataset for fall detection called (Qm1 fall dataset) containing 43 videos with 60fbs between 7 to 14 seconds, this is different from publicly available datasets. The proposed dataset encompasses a wide range of diverse and complex scenes. the living room and sitting room. Moreover, all fall events are real-world incidents rather than simulated fall events.

e. The model was trained and tested using three datasets, the first of which we created, and the second and third being public datasets.

Related Work

There are many studies about human fall detection. This research takes place in two ways: one using the devices a person wears, and the second using deep learning in surveillance cameras. In the first category, wearable devices are used [15, 16]. Kiprijanovska et al. [17] found fall detection using a wrist-worn device and deep learning method; they used long short-term memory and a set of 18 sensors to record and evaluate the data. Xin Yi et al. [16] used intelligent wearable devices. The device uses WIFI and BLE wireless, heart rate, oxygen sensors, and voice recognition. The sensor gets information and sends it to the WeChat application to know the situation and health status of older adults remotely. The second category can detect falls in three ways using multiple cameras based on local features [18] and using depth information [19]. The first way is to have multiple cameras record different angles of view. Kung et al. [20] used multiple cameras and a stream convolution neural network. The second way is to use local features. Wu et al. [21], used a deep multiple-instance learning framework with weak labels, their approach efficiently acquires knowledge of fall occurrences but requires detailed annotations. Detection outcomes are obtained by fusing information from two streams in a dual-modal network. Alanazi et al. [22] used multi-stream 3D CNN. The system assigns each stream to a particular phase, similar to human action recognition, which aims to identify various human actions, but it explicitly emphasises a four-branch design. A preprocessing procedure was conducted using the image-merging technique at two fusion levels, resulting in four sequentially merged images from 16 frames obtained from the input video. Li et al.[23] designed a dual-stream model using a convolution neural network, one for local features and other for external features; by enhancement of the transformer, this feature will feed the convolution neural network, allowing the streams to be merged to classify them for fall detection.

Chen et al. [24] presents a new FA-Fall model for detecting falls using ultrasonic and regular signals. This model uses passive audio signals from transmitting and receiving sensors. A noisy environment certainly affects the operation of the model. Mobsite et al. [25] proposed a model to learn and extract such features permanently without additional inputs. Fall-LSTM comprises a CNN-LSTM framework and two excitation modules: the Spatial Attention Module (SAM) and the Temporal Location Module (TLM). SAM provides spatial constraints on motion for feature layers through foreground extraction and spatial pooling. TLM emphasises frames with a high probability of falling events to LSTM by inferring the rate and trend of motion in the clips.

Kolobe et al. [26] employed a convolutional neural network to detect falls by extracting significant facial expressions from video frames. In this process, an image undergoes a series of processing steps, including applying the SoftMax activation function, after passing through many layers, such as the fully connected and pooling layers. The purpose of the pooling layer is to minimise the size of the feature maps by applying a sliding filter. Salimi et al. [27] used a convolutional neural network with LSTM and a 1-D convolutional neural network to find falls in video images of people falling or not falling based on pose estimation. They trained the model on the position of the human skeleton during all activities, then used the image to watch the person fall.

Many other studies have used deep learning for human fall detection. In other studies, deep learning and wearable sensors for fall detection were used, and most of these methods use a pressure sensor, an acceleration sensor, and a gyroscope. Yao et al. [28] presented a novel unsupervised fall-detection model comprising a feature extractor and predictor. Initially, they employed 3-D convolution and 3-D transposed convolution to create a feature extractor that captured the range–velocity–time characteristics of radar signals. Next, they created a predictor to analyze and understand the pattern of actions that do not include falling.

Chan et al. [29] proposed a model that combines convolutional LSTM (Conv-LSTM) networks with lightweight 3-D-CNN. Their model introduces a compact 3-D convolutional neural network.

Table 1. Datasets for Detecting Falls Based on Vision.

Dataset	Camera Type	Subjects	Fall Type	Other Activities	Trials	Variants	ML Method	Performance
D1&D2 [30]	2 RGB Cameras	4	Fall to the ground's surface	Standing, sitting, lying bending	4 actions		C-SVM, NN, KNN, RF, DT	Accuracy: 97.92% 97.84Precision %
UCF11, HMDB51, MCFD, and URFD [29]	2 cameras	51	Standing, sitting in a chair	Walking lying, sitting down	62 sequences		3D-CNN, LSTM	Accuracy: 97.28% Sensitivity: 100% Specificity: 99.15%
UP -Fall [31]	2 cameras	17	Backward, sideways, sitting, forward with hands, forward with knees	Sitting, walking, picking something up, standing, lying, jumping	3 receptions		SVM K-nearest neighbour Random forest NN	Precision: 13.04% Accuracy: 31.96% Recall: 13.73, specificity: 72.05%, F1-Score: 12.68%
TST v2 [32]	1 Microsoft Kinect v2	11	From the front, the back, and the side	Sitting down, picking something up off the ground, moving	264 sequences		Frame in depth	
AzurePose [33]	3 Microsoft Kintect v2	66		Drinking, shaking hands, waving hands	6 Sequences	10 action interactions	Regression RNN, Regression SVM, Bidirectional LSTM,	F1-Score: 52.6% 13.1% 33.3% 31.6%

NTU RGB-D [34]	3 Microsoft Kintect v2	Multiple actors	Fall to the floor	120 activities	Taken from 155 distinct camera angles	S-T Attention LSTM TM RNN S-T LSTM Part-Aware LSTM FSNet CNN	Accuracy: 57.9% 26.3% 44.9% 61.8% 62.4%

Consisting of five layers to mitigate the issue of overfitting. Adding channel- and spatial-wise attention modules to each layer improves the detection performance by letting the discriminatory features be examined in more detail. Furthermore, ConvLSTM is introduced to extract the extended spatial–temporal characteristics of 3-D tensors Kim et al. [35] proposed a sensor-hybrid deep learning model using an FNO IMU sensor and a gyroscopic sensor in a smartphone to predict the next movement of a human using the distance histogram and then convert the sensor datagram into 2-D image data. The model also used a Fourier neural operator to predict the fall of a human in this model wearing the device quickly for danger based on the Fourier neural operator. Table 1 provides an overview of many datasets for fall detection and other actions, including the number of cameras and some results. Lazzi et al. [30] constructed a model that has a first phase for finding silhouettes, a second phase for feature extraction, and a third phase for classifying using the D1and D2 dataset. Martínez et al. [31] used the UP-Fall dataset with two cameras to detect body movements, such as sitting, moving sideways, and walking. They used NN, random forest, SVM, and K-nearest neighbour, all other details being provided in the table. Gasparrini et al. [32] used the TST v2 dataset with one Microsoft Kinect v2 camera to detect human action from different angles, such as picking something up, sitting down, and falling.Bull et al [33] used the Azure Pose dataset with three Microsoft Kinect v2 cameras to detect actions, such as drinking, waving hands, and shaking hands; they used machine learning with S-T Attention LSTM, Bidirectional LSTM, and regression SVM. Liu et al. [34] used the NTU RGB-D dataset with three Microsoft cameras to detect many actions, such as walking, running, and jumping, using Part-aware LSTM, RNN, and S-T LSTM, with other details provided in the table.

Methodology

We proposed a novel deep-learning approach that combines Inception v3 and LSTM in a hybrid model in the last layers of the model to enhance the classification accuracy. The initial step is preprocessing a data stage for training and testing. Standardisation procedures include resizing images to 224 × 224 pixels, transforming RGB colour values into grayscale colour, subtracting the background, and normalising edge detection before model training.

Google TensorFlow [36] was employed to implement the deep-learning methods in this study. It provided high-level APL and did not need to prepare a neural network or program because complex code solves all these problems.

Pre-Processing Dataset

The pre-processing dataset is a crucial step in the model. It influences the model's accuracy, performance, and efficiency by deleting features that are undesirable or do not help the model detect the human posture and maintain the feature that the model needs to increase its efficiency and accuracy.

Resizing Images to 224 × 224 Pixels. Padding can prevent distortion and keep the original proportions when scaling pictures to 224 × 224 pixels to maintain the correct aspect ratio. This is

crucial when preparing photos for machine learning models, especially those created for image-recognition tasks. This technique may be carried out by using one of many image-processing packages.

The photos must be resized to 224 × 224 pixels to be compatible with various machine-learning models (see Figure 1). This uniform size enables consistent input dimensions for various models.

Figure 1. Renovating image dimensions to 244 × 244 pixels.

Converting Colour to Grayscale. The second step in the pre-processing is to convert the image into grayscale to eliminate unnecessary information from the coloured/RGB images, as they include an abundance of data that may not be necessary for further processing. Converting photos to grayscale involves discarding unnecessary information. This process is necessary to increase the accuracy and efficiency of the model.

After converting to grayscale, grey extracts the grayscale between adjacent cells, which describes the proportion of each grayscale in the image's pixels, as shown in formula (1).

$$H(i) = \frac{n_i}{N} \; i = 0,1, \ldots L-1 \qquad (1)$$

where i represents the grey level of the pixel, N the number of pixels in the image, n_i the pixels in the grey level, and L the total number of grey pixels.
To efficiently extract the grey features, the following statistics are chosen:

a. Mean: The average value of the pixels in the image, illustrated in formula (2).

$$\mu = \sum_{i=0}^{L-1} iH(i) \qquad (2)$$

b. Variance: This measures the extent of pixel scattering. As the value increases, the distribution becomes increasingly spread out, as shown in formula (3).

$$\delta^2 = \sum_{i=0}^{L-1} (i-\mu)^2 H(i) \qquad (3)$$

c. Peak: This is used to determine whether the image is in an elevated or level position, as depicted in formula (4).

$$\mu_k = \frac{1}{\delta^4} \sum_{i=0}^{L-1} (i-\mu)^4 H(i) - 3 \qquad (4)$$

Background Subtraction. This stage is crucial for enhancing the precision of the model. The images' primary objective was to emphasise the specific descent without requiring the surroundings. It is crucial to remove the backdrop to do this, which the 'rembg' package does. By using image filtering, the accuracy of edge detection is improved

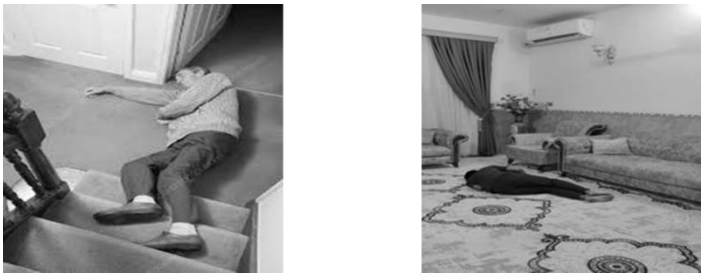

Figure 2. Converting images to grayscale.

Figure 3. Deleting the background.

Edge Detection for Images. Image normalisation is a common pre-processing step in computer-vision and machine-learning tasks. It involves transforming the pixel values of an image into a standardised range to improve model performance and convergence during training. The smoothed image undergoes filtering using Sobel kernels in both the vertical and horizontal directions to yield the first derivative in the vertical (E_y) and horizontal (E_x) directions. The edge advancement and direction for each pixel can be ascertained from these two photos, as shown in formula (5).

$$Edge\ Gradient(E) = \sqrt{E_x^2 + E_y^2} \tag{5}$$

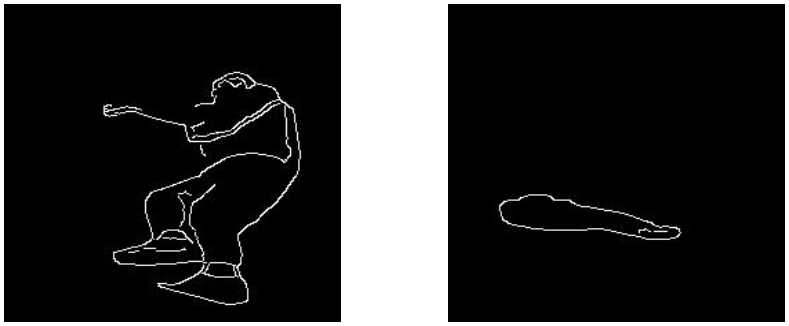

Figure 4. Edge detection for images.

LSTM

LSTM is similar to Recurrent Neural Network (RNN) which offers a higher level of complexity and specificity. LSTM outperforms RNN in terms of internal memory capacity and is specifically engineered for handling sequences [37]. In this study, the LSTM model receives the outputs from the Inception v3 model, which classifies the postures during falling. The LSTM model requires a sequential input of data for both training and testing purposes. Prior to entering the LSTM, it is necessary to collect consecutive data. For instance, if the arranged frame includes the subsequent sequence of postures without any falls, the output of the LSTM should show that this movement is considered normal. It contain the input gate, output gate, and forget gate of the framework of LSTM, as shown in Figure 5. LSTM has a new input, x_t, output, h_t, and forget input, f_t.

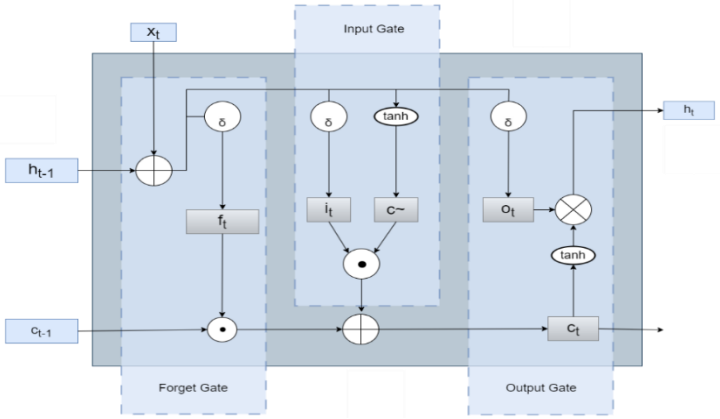

Figure 5. LSTM memory cell diagram.

In Figure 5, the symbol 'Θ' denotes the multiplication of vector elements, while the symbol '□' indicates the sum of the vectors. The variables h_t, f_t, o_t, c_t, i_t, and c~ correspond to the output results of the forget gate, input gate, input node, output gate, memory-unit state, and implicit state, respectively.

When adding LSTM to Inception v3, one must delete the SoftMax layer because it is used for classification and probability, but our proposed model used sequence classification; this domain of LSTM and the design of LSTM can process sequence data by removing the SoftMax layer, which can feed the data from Inception v3 to LSTM directly to regression or classifying it as a fall or normal.

Enhancement Inception v3 Model

Inception v3 is a convolutional neural network (CNN) developed by Google specifically for picture-classification jobs. The Inception architecture's third iteration was developed by Szegedy et al [38]. in 2015. Inception v3 is a successor to the previous Inception architectures, aiming to enhance the performance and efficiency of image-categorisation tasks. The Inception v3 architecture is characterised by its depth and complexity, comprising a series of interconnected inception modules. An inception module is a combination of various convolutional and pooling layers that are specifically designed to extract distinct characteristics from the input image.

Detecting falls in humans faces several obstacles, including overfitting, because most databases used for training take an image of a person with all the furniture in the house, which causes overfitting. Srivastava et al. [39] proposed a method to reduce overfitting by freezing units of the neural network based on specific calculations, and this method significantly reduced overfitting. The steps of this method are:

1) Select neurons randomly and deactivate them without altering the input or output.

2) Backpropagation is still used to compute the network's losseven if the input is delivered to the network for forward propagation. In the absence of forward propagation of the training data and without executing the backpropagation of the error gradients are finished, the parameters of the dormant neurons remain unchanged, but the parameters of the active neurons are optimised.

3) This procedure is repeated until the loss function is stable.

The dropout of this algorithm is counted by formulas (7) and (8):

$$r_j^{(l)} \sim Bernoulli(p) \qquad (6)$$

$$y_i^{l+1} = f\left(w_i^{l+1}.I(x).r_j^l + p_i^{l+1}\right) \qquad (7)$$

Where $r_j^{(l)}$ is a possibility vector holding in just 0 and 1 bits and agree a Bernoulli distribution,

p_i^{l+1} and w_i^{l+1}represent the bias and weight of the next layer, and I(x) symbolises the extracted feature. y_i^{l+1}Indicates the output after progress through the activation function. The dropout of the model is depicted in Figure 6.

Inception v3 created four new structures, leaving the original classification and keeping one assistant classification in the middle. The Inception model A classifier has achieved depth and efficiency in data analysis, as it uses, instead of the original 5*5 convolution kernel, two 3*3 convolution kernels, whereas the second convolutional layer uses the same 5*5 parameters to reduce the number of variables in the network significantly**.**

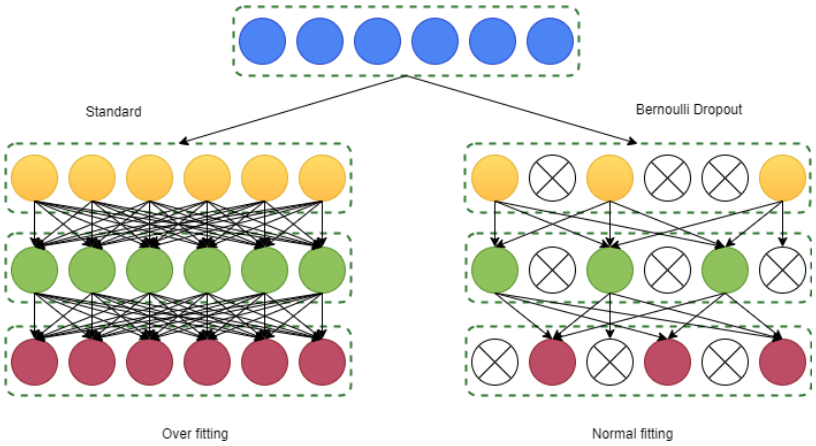

Figure 6. Dropout in Bernoulli.

To count the number of parameters, formulas (9) and (10) were used:

$$O_p = (H \times W \times C) \times (5 \times 5 \times C) = 25HWC^2 \quad (8)$$

$$R_p = 2(H \times W \times C) \times (3 \times 3 \times C) = 18HWC^2 \quad (9)$$

Where O_p is the original parameter for convolutional kernels, $R_p$the replacement parameter, H the height of the image, W the width of the image, and C the number of filters. Note from formulas (3) and (4) that the number of parameters are reduced by about 28%. In model B, instead of using an N × N convolution kernel, it used an N × 1 and a 1 × N to decrease the total number of parameters and enhance the computational efficiency. In Inception Module C, the approach of growing horizontally compared to vertically is used, leading to an augmented representation dimension and a higher feature dimension.Hence, to address the issue of overfitting and enhancing recognition accuracy, this paper suggests an enhanced Inception v3 model that integrates the strengths of the original model, such as its powerful expressive capacity and abundant spatial features, along with the effective overfitting prevention capability of dropout.

Figure 6 shows the full architecture of the model for fall detection. We assume the set of training in the input is g in this paper. The size of the vector feature in input size is $X(n_x,g)$, the sample of the input data is $X \in R^{n_x \times g}$, and the output data is progress $Y \in R^{1 \times g}$.. For example, we take layer m in the model to process the forward propagation in formulas (11) and (12):

$$Z^m = w^m . A^{[m-1]} + b^m \quad (10)$$

$$A^m = R^m(Z^m) \quad (11)$$

Where A^mrepresents the input to the to layer m, b^mrepresents the bias of layer, w^mrepresents the weight of the layer, and R^mrepresents the output using a nonlinear activation function using ReLU. The formulas are as stated:

$$Cost = -\frac{1}{g}\sum_{i=1}^{g}\left(y^{i}\, log\left(A^{[m](i)}\right) + \left(1 - y^{i}\right) log\left(1 - A^{[m](i)}\right)\right) \quad (12)$$

$$Z^{m} = dA^{m} \times R^{m'}(Z^{m}) \quad (13)$$

$$dw^{m} = \frac{1}{g} dZ^{m}.A^{[m-1]T} \quad (14)$$

$$db^{[m]} = \frac{1}{g}\sum_{i=1}^{g} b^{[m](i)} \quad (15)$$

After obtaining dw^{m} and $db^{[m]}$, we will update the parameters using the learning rate, and the above process will continuously repeat until the training model process is complete.

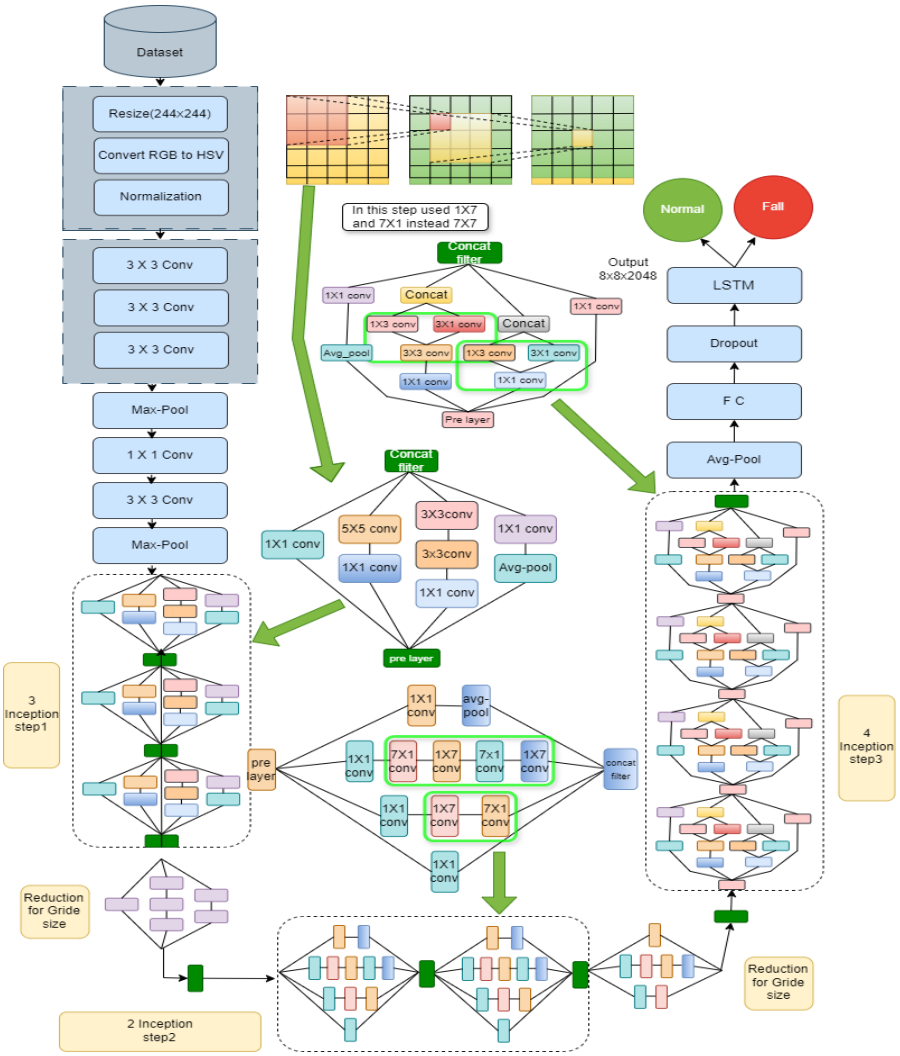

Figure 7. Proposed model structure.

Hence, to address the overfitting problem caused by an insufficient sample size of the fall-detection dataset and enhance the detection accuracy, this study proposes the improved Inception v3 model. This model combines the strengths of Inception v3, such as its strong expressive ability and abundant spatial features that facilitate parallel convolution, with the strong ability to prevent overfitting through dropout. Figure 7 shows the improved architecture of Inception v3.

Training Model

The training set contains images of falls and non-falls fed into the trained neural network model. The model uses convolution in the initial layers to extract shallow features from the input images. These features are then pooled to reduce the size of the feature map. The Inception block is then used to extract deep features from the pooled features. The Inception block uses parallel

convolution layers to extract varying-size feature maps. These feature maps are subsequently merged after each block of Inception and fed into the subsequent layer.

Next, the features extracted are fed into the global average pooling layer, followed by the addition of a dropout layer. When the global average pooling layer results pass through the dropout layer, they are input into the LSTM. Specific neuron nodes are randomly ignored in the dropout layer at a specific ratio. It is important to note that the input and output layers remain unchanged throughout this process. The LSTM layer generates the classification results, finalising the forward-propagation process. Once the forward propagation is finished, the parameters in Inception v3 are updated using the backpropagation technique, excluding the ignored neuron nodes; this completes an iterative process. The processes mentioned above are iterated until the conclusion of the training procedure. Moreover, to implement this strategy, we removed the SoftMax layer and added LSTM to classify image which we got from the video the doing well and increase accuracy. Figure 8 depicts the training procedure for the model.

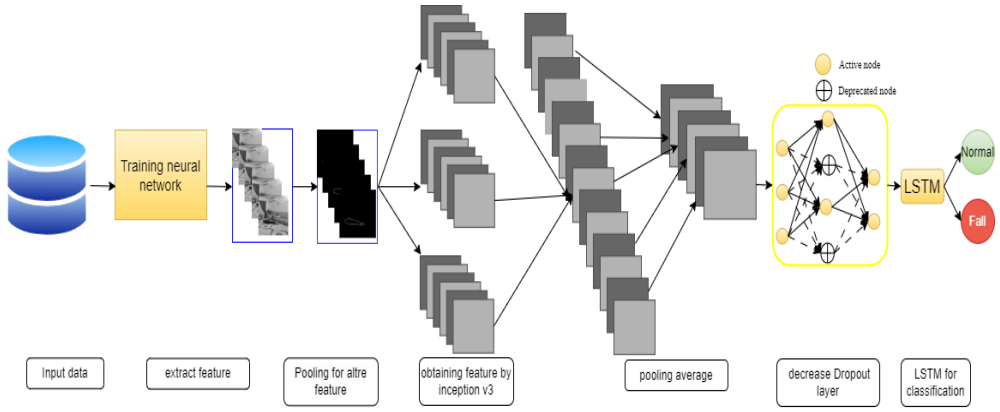

Figure 8. Training model with Inception v3 and LSTM.

The fall-detection model Inception V3_LSTM primarily comprises five components: data preprocessing, the Inception v3 layer, the pooling layer, the LSTM optimisation layer (see Figure 9), and the output layer. The algorithm model presented in this paper differs from the conventional fall-detection paradigm. The fusion approach achieves excellent detection accuracy because of Inception v3's ability to automatically extract features and the LSTM network's inclusion of storage units in the hidden layer, which can effectively capture long-term correlations in time series data.

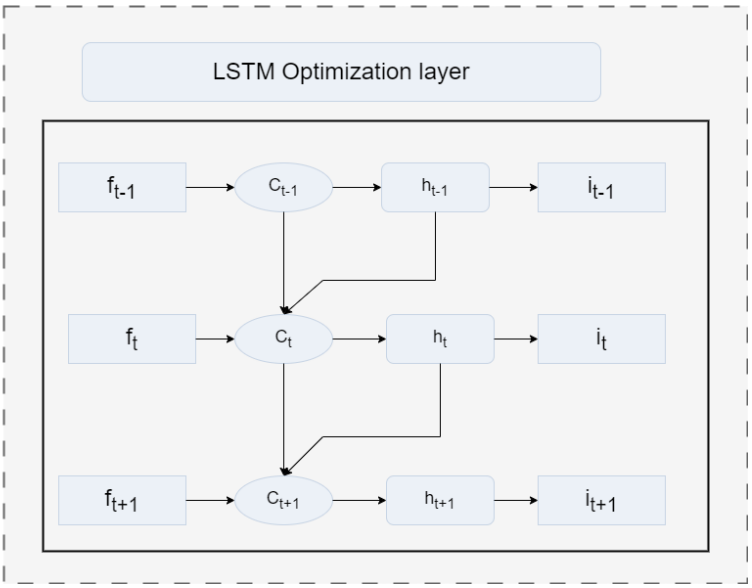

Figure 9. LSTM optimization layer.

The output layer uses the LSTM classifier to merge the data processed by the LSTM optimisation layer to carry out unnormalised probability calculation. This is specifically used for the binary classification problem of fall detection. Determine the expected probability of falling among daily activities and the results of the classification of daily activities.

Dataset

This study's methodology is based on the use of three datasets. Before classifying each human image as either training or test data, we perform a thorough evaluation to confirm its quality. Through comprehensive evaluation of each photograph, we ensure the maintenance of rigorous standards and precision in our research methodology.

The first dataset is the Qm1 fall dataset. We created a new dataset for fall detection using a single camera containing 43 videos in two groups, sitting rooms and sleeping rooms, with 60fbs, between 7 to 14 seconds, which differs from the existing publicly available datasets, as it covers most of the places where the elderly are present. The proposed dataset contains diverse and complex scenes. Moreover, all fall events are real-world incidents rather than simulated fall events. This database distinguishes itself from others by emphasising home areas and the diverse ambiance they elicit. This database categorises videos according to the specific room in which they were filmed, providing a detailed examination of how various settings might affect the mood and subject matter of the videos. This distinctive characteristic distinguishes it from broader video collections and renders it a significant asset for individuals interested in investigating the importance of human fall detection.

The second public dataset is called the fall-detection dataset [40] and contains two groups: training 374 images and validating 111 images.

The third dataset, The Le2i dataset [41], was used in this study in two groups: home_01 and home_02 contain 30 videos each. After preprocessing, their size becomes 224 × 224. To assess their resilience, they captured video footage of the dataset from various settings, including home.

Result and Discussion

The results of a training phase using a model for classifying photos of Fall or No-Fall. Table 2 includes information on the loss, accuracy, implementation time, and other details for each phase of a sample training run. The loss values indicate the difference between the predicted and actual labels, with lower values indicating better performance.

Table 2. Performance Evaluation Metrics of Fall-Detection Models on Benchmark Datasets (Fall-Detection, Le2i, and Qm1).

Metric	Fall-Detection Dataset	Le2i Dataset	Qm1 Fall Dataset
Accuracy	0.9619	0.97	0.9800
Precision	0.9628	0.96	0.9800
Recall Score	0.9609	0.96	0.9763
F1-Score	0.9592	0.97	0.9788
MSE	0.0400	0.02	0.0200
RMSE	0.1952	0.18	0.1414

According to Table 3, the accuracy score of the model was 0.98 in our dataset, 96% in the fall-detection dataset, and 97% for the Le2i dataset. Similarly, the recall score of 0.97 indicates that the model correctly identified 98% of the positive cases. Figure 11 shows the loss function for the model decreasing at the end of model training in the Qm1 fall dataset, and Figure 10 shows the increasing accuracy after each training period. Figure 13 and Figure 12 show the curve of loss function and accuracy, respectively, in the fall-detection dataset. Note that the accuracy will

increase after each phase and loss of model stability. Figure 17 shows the accuracy of the model in the Le2i dataset.

The effectiveness of the proposed classification model in accurately distinguishing between instances of falls and non-falls within the dataset is demonstrated through the evaluation of categorisation performance, as depicted in Figure 12, which shows the loss function for our dataset, and Figure 13, which shows the accuracy for our dataset. The values of the matrices represent the accuracy of predictions for both fall and non-fall events.

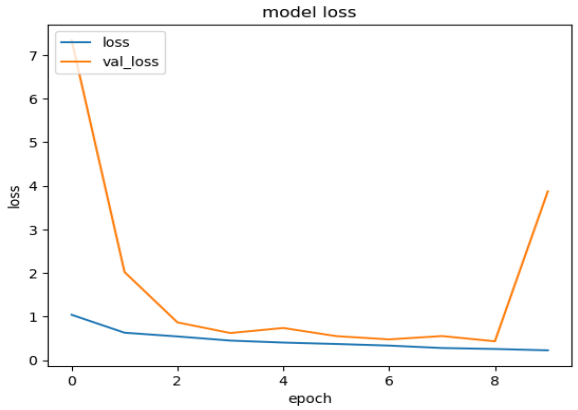

Figure 10. Loss function for model on fall-detection dataset.

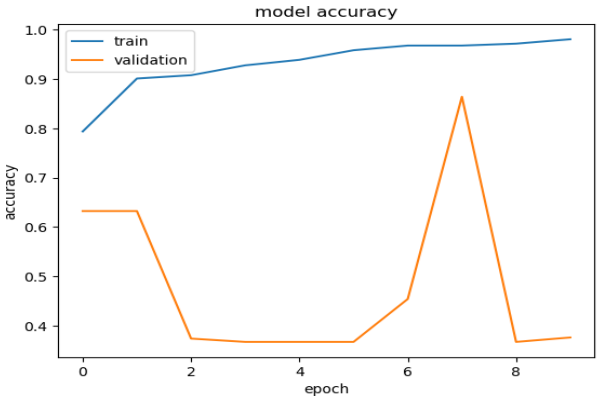

Figure 11. Accuracy for Qm1 dataset.

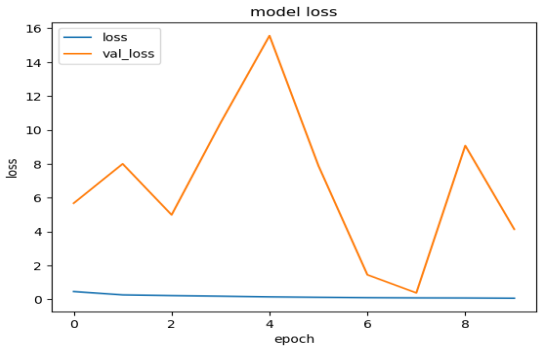

Figure 12. Loss function for Qm1 dataset.

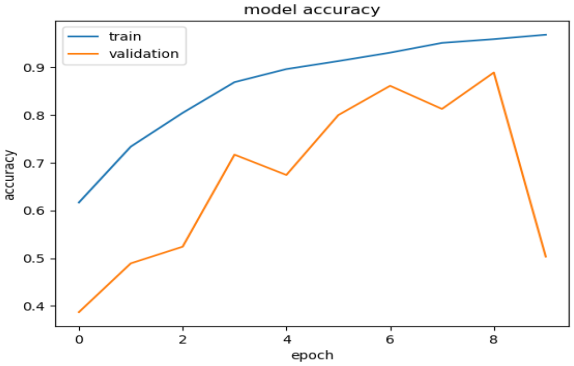

Figure 13. Accuracy for model on fall-detection dataset.

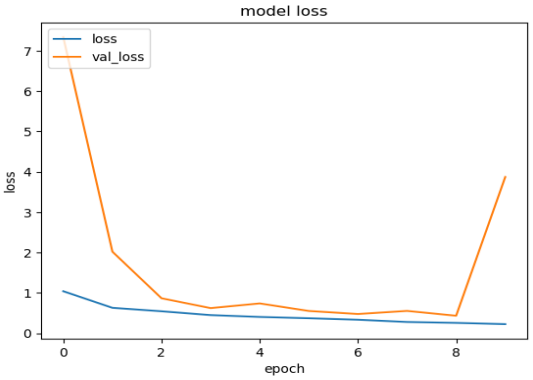

Figure 14. Accuracy for model loss and epochs.

The provided metrics are related to the performance evaluation of a fall-detection method that combines Inception v3 and Support Vector Machines (LSTM) for classification. This method involves data augmentation, a technique used to artificially increase the diversity and size of a dataset by generating additional training examples through various manipulations.

The results for the fall-detection method using classification with data augmentation for the Qm1 fall dataset are:

- Accuracy: 0.98
- Precision: 0.98
- Recall Score: 0.9762618690654672
- F1 Score: 0.9788014718582996
- Mean Squared Error: 0.02
- Root Mean Squared Error (RMSE): 0.1414213562373095

The results for the Le2i Dataset using our model are:

- Accuracy: 0.9772535805
- Precision: 0.9661921708
- Recall Score: 0.9696882161000726
- F1 Score: 0.9757412399
- Mean Squared Error: 0.02
- RMSE: 0.1843873386465664

The results for the fall-detection dataset using our model are:

- Accuracy: 0. 96
- Precision: 0. 9609
- Recall Score: 0.9654819329022582
- F1 Score: 0.9592425115522564
- Mean Squared Error: 0.04
- Root Mean Squared Error (RMSE): 0.1952127829935147

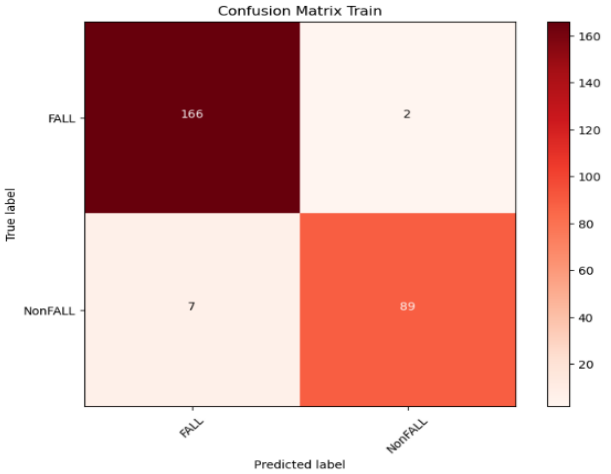

Figure 14. Confusion matrix for fall detection dataset.

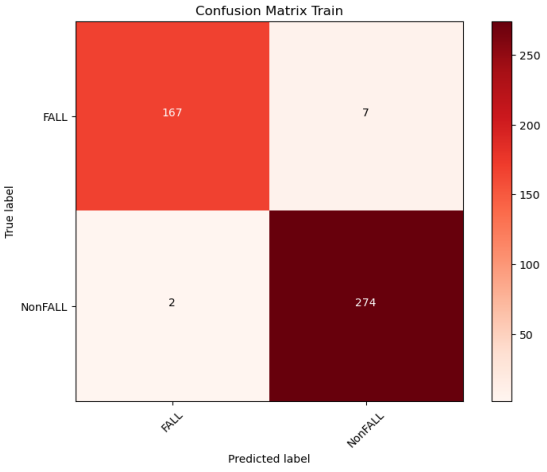

Figure 15. Confusion matrix for Qm1 fall dataset

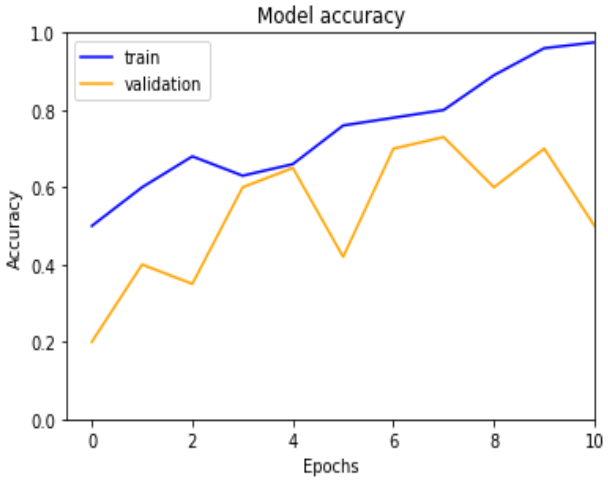

Figure 16. Accuracy for fall-detection Le2i.

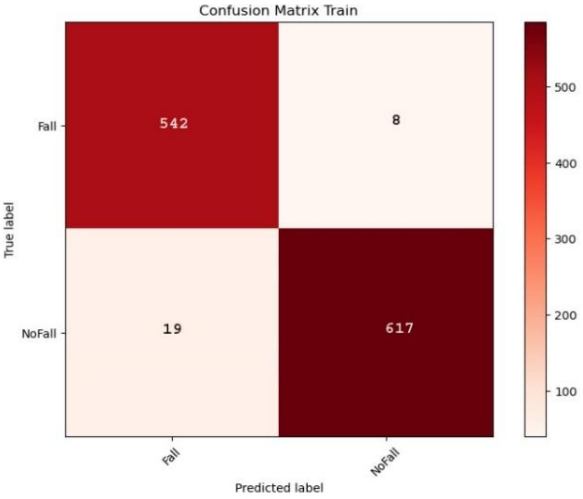

Figure 17. Confusion matrix for Le2i dataset.

Symmetry reflects the performance of the fall-detection system. The high accuracy, precision, recall score, and F1 score values suggest that the system effectively identifies fall instances. In Figure 16, the Result test image in our dataset, and the Result test image in the fall detection dataset. The low mean squared error and RMSE values indicate that the predicted fall labels closely align with the actual fall labels, reinforcing the accuracy of the system's predictions. The inclusion of data augmentation likely contributes to improved performance by providing a more diverse and representative training dataset for the FDFE model.

Evaluating Against the Current Highest Level of Achievement

Table 3 compares our proposed model with other state-of-the-art techniques. We evaluated our method by comparing it with previous state-of-the-art techniques on the Le2i fall-detection dataset. The comparison methods exclusively analyzed RGB videos without incorporating supplementary inputs, such as depth images or sensor data—previous studies on Le2i commonly computed assessment metrics like accuracy, specificity, and sensitivity. Figure 16 shows a simple result model from our dataset and fall-detection dataset with the expectation rate for each image.

Table 3. Comparison of our Proposed Model with Other State-of-the-Art Methods.

Method	Accuracy	Specificity	Sensitivity
Song Zou et al. [42]	97.2	97	100
J. Thummala [43]	95.2		
Adrian et al.[44]	97.0	97.0	99.0
Wang et al. [45]	96.9	96.5	97.4
Dentamaro et al.[46]	89.0	88.0	89.0
Inception v3-LSTM	97.7	97.	98.5

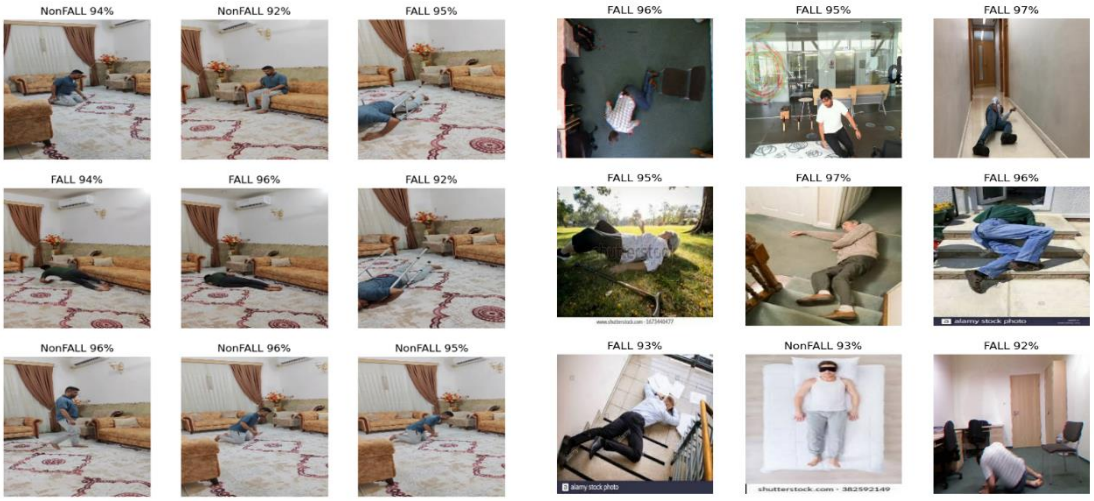

Figure 18. Sample of resulting model.

Conclusion

This study presents a hybrid model that uses Inception v3 and LSTM algorithms to accurate identify all events in older inaccurately individuals' homes. The research involved analyzing three datasets specifically for this purpose. The automatic detection model suggested in this paper primarily focuses on enhancing the following elements:

(1) We successfully mitigated the issue of overfitting during the model's learning process, where we added a dropout layer to the Inception v3 model after the average pooling layer;

(2) There are two advantages to training the model with this the datasets and transfer learning integrated: firstly, it accelerates the model training pace, and, secondly, it helps to mitigate the problem of overfitting when the amount of data is significant.

We used three datasets to train and test the model: the first dataset, our dataset, the second fall-detection dataset, and the third Le2i dataset.

The results suggest that this approach can significantly contribute to developing automated fall-detection systems, which can have critical applications in various domains, including healthcare, eldercare, and safety monitoring. Further research and experimentation can explore ways to improve the model's performance, adapt it to specific environments or populations, and address any limitations or challenges that may arise in its real-world deployment.

- Funding: This research was funded by the first author.
- Data Availability Statement: The data set will be available upon request.
- Conflicts of Interest: I declare that there is no conflict of interest.

References

[1] Bektan Kanat, B. and O. Incealtin, Comparison of Risk Factors for Falls in the Old and the Oldest Old Admitted to the Emergency Department. Harran Üniversitesi Tıp Fakültesi Dergisi, 2023.

[2] Bull, F.C., et al., World Health Organization 2020 guidelines on physical activity and sedentary behaviour. British Journal of Sports Medicine, 2020. 54: p. 1451 - 1462.

[3] Hirooka, K., et al., Ensembled Transfer Learning Based Multi-channel Attention Networks for Human Activity Recognition in Still Images. IEEE Access, 2022. PP: p. 1-1.

[4] Khan, I.U., S. Afzal, and J.-W. Lee, Human Activity Recognition via Hybrid Deep Learning Based Model. Sensors (Basel, Switzerland), 2022. 22.

[5] Hayat, A., et al., Human Activity Recognition for Elderly People Using Machine and Deep Learning Approaches. Inf., 2022. 13: p. 275.

[6] Sadreazami, H., M. Bolic, and S. Rajan, Contactless Fall Detection Using Time-Frequency

Analysis and Convolutional Neural Networks. IEEE Transactions on Industrial Informatics, 2021. 17: p. 6842-6851.
[7] Cun, W., et al., Sitting posture detection and recognition of aircraft passengers using machine learning. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2021. 35: p. 284 - 294.
[8] Randhawa, P., et al., Human activity detection using machine learning methods from wearable sensors. Sensor Review, 2020. 40: p. 591-603.
[9] Wu, Q., et al., Real-time running detection system for UAV imagery based on optical flow and deep convolutional networks. IET Intelligent Transport Systems, 2020.
[10] Straczkiewicz, M., E.J. Huang, and J.-P. Onnela, A "one-size-fits-most" walking recognition method for smartphones, smartwatches, and wearable accelerometers. NPJ Digital Medicine, 2022. 6.
[11] Wu, C., et al., Video Anomaly Detection using Pre-Trained Deep Convolutional Neural Nets and Context Mining. 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), 2020: p. 1-8.
[12] Khalili, S., H. Mohammadzade, and M.M. Ahmadi, Elderly Fall Detection Using CCTV Cameras under Partial Occlusion of the Subjects Body. ArXiv, 2022. abs/2208.07291.
[13] Juraev, S., et al., Exploring Human Pose Estimation and the Usage of Synthetic Data for Elderly Fall Detection in Real-World Surveillance. IEEE Access, 2022. 10: p. 94249-94261.
[14] Yu, Z., et al., An Elderly Fall Detection Method Based on Federated Learning and Extreme Learning Machine (Fed-ELM). IEEE Access, 2022. 10: p. 130816-130824.
[15] Yhdego, H., et al., Wearable Sensor Gait Analysis of Fall Detection using Attention Network. 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021: p. 3137-3141.
[16] Yi, X., et al., Design of Intelligent Wearable Devices for the Elderly Based on ARM. 2023 3rd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), 2023: p. 263-266.
[17] Kiprijanovska, I., H. Gjoreski, and M. Gams, Detection of Gait Abnormalities for Fall Risk Assessment Using Wrist-Worn Inertial Sensors and Deep Learning. Sensors (Basel, Switzerland), 2020. 20.
[18] Maldonado-Mendez, C., et al., Fall detection using features extracted from skeletal joints and SVM: Preliminary results. Multimedia Tools and Applications, 2022. 81: p. 27657 - 27681.
[19] Gutiérrez, J., V. Rodríguez, and S. Martín, Comprehensive Review of Vision-Based Fall Detection Systems. Sensors (Basel, Switzerland), 2021. 21.
[20] Kong, Y., et al., Learning spatiotemporal representations for human fall detection in surveillance video. J. Vis. Commun. Image Represent., 2019. 59: p. 215-230.
[21] Wu, L., et al., Robust fall detection in video surveillance based on weakly supervised learning. Neural networks : the official journal of the International Neural Network Society, 2023. 163: p. 286-297.
[22] Alanazi, T.O. and G. Muhammad, Human Fall Detection Using 3D Multi-Stream Convolutional Neural Networks with Fusion. Diagnostics, 2022. 12.
[23] Li, B., J. Li, and P. Wang, Fall detection algorithm based on global and local feature extraction. Pattern Recognition Letters, 2024.
[24] Chen, D., A. B. Wong, and K. Wu, Fall Detection Based on Fusion of Passive and Active Acoustic Sensing. IEEE Internet of Things Journal, 2024. 11: p. 11566-11578.
[25] Mobsite, S., et al., A Deep Learning Dual-Stream Framework for Fall Detection. 2023 International Wireless Communications and Mobile Computing (IWCMC), 2023: p. 1226-1231.
[26] Kolobe, T.C., C. Tu, and P.A. Owolawi, Fall recognition system using convolutional neural network. 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2022: p. 1-6.
[27] Salimi, M., J.J.M. Machado, and J.M.R.S. Tavares, Using Deep Neural Networks for Human Fall Detection Based on Pose Estimation. Sensors (Basel, Switzerland), 2022. 22.

[28] Yao, Y., et al., Unsupervised-Learning-Based Unobtrusive Fall Detection Using FMCW Radar. IEEE Internet of Things Journal, 2024. 11: p. 5078-5089.
[29] Su, C., et al., A novel model for fall detection and action recognition combined lightweight 3D-CNN and convolutional LSTM networks. Pattern Analysis and Applications, 2024. 27: p. 1-16.
[30] Iazzi, A., M. Rziza, and R.O.H. Thami, Fall Detection System-Based Posture-Recognition for Indoor Environments. Journal of Imaging, 2021. 7.
[31] Martínez-Villaseñor, M.d.L., et al., UP-Fall Detection Dataset: A Multimodal Approach. Sensors (Basel, Switzerland), 2019. 19.
[32] Gasparrini, S., et al. Proposal and Experimental Evaluation of Fall Detection Solution Based on Wearable and Depth Data Fusion. in ICT Innovations. 2015.
[33] Bashirov, R., et al., Real-time RGBD-based Extended Body Pose Estimation. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 2021: p. 2806-2815.
[34] Liu, J., et al., NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. 42: p. 2684-2701.
[35] Kim, T., et al., Predicting Human Motion Signals Using Modern Deep Learning Techniques and Smartphone Sensors. Sensors (Basel, Switzerland), 2021. 21.
[36] Abadi, M., et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. ArXiv, 2016. abs/1603.04467.
[37] Sherstinsky, A., Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. ArXiv, 2018. abs/1808.03314.
[38] Szegedy, C., et al., Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: p. 2818-2826.
[39] Srivastava, N., et al., Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 2014. 15: p. 1929-1958.
[40] Kandagatla, U.K., Fall Detection Dataset. 2021, Kaggle.
[41] Charfi, I., et al., Optimised spatio-temporal descriptors for real-time fall detection : comparison of SVM and Adaboost based classification. Journal of Electronic Imaging, 2013. 22: p. 17.
[42] Zou, S., et al., Movement Tube Detection Network Integrating 3D CNN and Object Detection Framework to Detect Fall. Electronics, 2021. 10: p. 898.
[43] Thummala, J. and S. Pumrin, Fall Detection using Motion History Image and Shape Deformation. 2020 8th International Electrical Engineering Congress (iEECON), 2020: p. 1-4.
[44] Núñez-Marcos, A., G. Azkune, and I. Arganda-Carreras, Vision-Based Fall Detection with Convolutional Neural Networks. Wirel. Commun. Mob. Comput., 2017. 2017.
[45] Wang, B.-H., et al., Fall Detection Based on Dual-Channel Feature Integration. IEEE Access, 2020. 8: p. 103443-103453.
[46] Dentamaro, V., D. Impedovo, and G. Pirlo, Fall Detection by Human Pose Estimation and Kinematic Theory. 2020 25th International Conference on Pattern Recognition (ICPR), 2021: p. 2328-2335.