



Article

Classification of Indoor Human Fall Events Using Deep Learning

Arifa Sultana ¹, Kaushik Deb ^{1,*} , Pranab Kumar Dhar ¹ and Takeshi Koshiba ² 

- ¹ Department of Computer Science and Engineering, Chittagong University of Engineering & Technology (CUET), Chattogram 4349, Bangladesh; arifa.z@eastdelta.edu.bd (A.S.); pranabdhar81@cuet.ac.bd (P.K.D.)
- ² Faculty of Education and Integrated Arts and Sciences, Waseda University, 1-6-1 Nishiwaseda, Shinjuku-ku, Tokyo 169-8050, Japan; tkoshiba@waseda.jp
- * Correspondence: debkaushik99@cuet.ac.bd

Abstract: Human fall identification can play a significant role in generating sensor based alarm systems, assisting physical therapists not only to reduce after fall effects but also to save human lives. Usually, elderly people suffer from various kinds of diseases and fall action is a very frequently occurring circumstance at this time for them. In this regard, this paper represents an architecture to classify fall events from others indoor natural activities of human beings. Video frame generator is applied to extract frame from video clips. Initially, a two dimensional convolutional neural network (2DCNN) model is proposed to extract features from video frames. Afterward, gated recurrent unit (GRU) network finds the temporal dependency of human movement. Binary cross-entropy loss function is calculated to update the attributes of the network like weights, learning rate to minimize the losses. Finally, sigmoid classifier is used for binary classification to detect human fall events. Experimental result shows that the proposed model obtains an accuracy of 99%, which outperforms other state-of-the-art models.



Citation: Sultana, A.; Deb, K.; Dhar, P.K.; Koshiba, T. Classification of Indoor Human Fall Events Using Deep Learning. *Entropy* **2021**, *23*, 328. <https://doi.org/10.3390/e23030328>

Academic Editor: Jose Santamaria Lopez

Received: 18 February 2021
Accepted: 4 March 2021
Published: 10 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: human fall classification; deep learning; recurrent neural network (RNN); convolutional neural network (CNN); gated recurrent unit (GRU)

1. Introduction

Human fall detection system is an important segment of assistive technology since living assistances are very much obligatory for many people. There has been a remarkable emersion in the elderly population in Bangladesh as well as in western countries over recent years. The statistics on human fall detection have exposed that falls play a key role in injurious death for elders more than 79 years of age [1]. According to the review from the National Institutes of Health of the United States, approximately 1.6 million aged people are sustaining fall-related excoriations in the U.S. every year [2]. Meanwhile, China is facing the fastest aging population in human history, the population will rise to about 35% by 2050 [2] from 2020. A study shows that about 93% of the elders among which 29% live alone in a house [3]. It was verified that about 50% of the aged people lying on the floor because of fall events for more than one hour, will die within six months even though they do not have any direct injuries [4].

Another statistic provided by the Public Health Agency of Canada [5] reported mentionable data. In 2026, one Canadian older than 65 will be out of five where in 2001 the portion was eight to one. It is notable that 93% of elderly people stay in their private house among which 29% of them lead a lonely life [5]. Again almost 62% of injury-related hospitalizations for elders are the result of falls [6].

To identify fall at right, in recent years various methods are proposed using advanced devices like wearable sensors, accelerometers, gyroscope, magnetometers and so on. However, this is not an effective solution since it is impractical to wear a device for a long time [7].

Therefore, it is indispensable to initiate a penetrating surveillance system for senior people, which can immediately and automatically detect fall actions inside the room and notify the state to the caretakers. This is only possible when a sensor-based alarm generation system is placed in the living room.

In this paper, a deep learning and vision-based framework is proposed for human fall detection and classification that can also monitor old people in the indoor environment incorporating a convolutional neural network (CNN) with the recurrent neural network (RNN). Among different types of RNN, gated recurrent unit is integrated along with CNN. We have also explored some transfer learning models like VGG16, VGG19 followed by GRU to classify human fall action. Besides, we have also assessed our proposed model using the two most prominent datasets, UR fall detection dataset and Multiple cameras fall dataset. Our proposed model shows an impressive performance using these datasets compared to other existing models.

Moreover, with the expansion of technology, human beings are leading a sedentary lifestyle which has numerous negative impacts [8]. Because of this lifestyle, people are suffering from many more diseases like chronic health disease, muscle weakness, labyrinthitis, osteoporosis. Besides, the number of lonely living beings is increasing with the advancement of technology which represents the necessity of a monitoring system to detect adverse events [9]. This model will be applicable in the medical alert system and even in the old home for monitoring individuals. However, the key contributions of our proposed model are enlisted below:

- Building a scratch model combining CNN with GRU to classify human fall events from daily living activities.
- Exploring a transfer learning approach through training some recent pre-trained models like Xception, VGG integrating with GRU and evaluating their performance.
- Fine-tuning two pre-trained models: VGG16 and VGG19 along with GRU.
- Assessing the performance of the proposed model with some deep learning models like 3DCNN, 1DCNN incorporating with GRU.
- Executing some best performing recurrent neural networks: Long and Short Term Memory (LSTM), Bidirectional LSTM (BI-LSTM) combining with the proposed 2DCNN network and evaluating performance over two challenging datasets, UR fall detection dataset and Multiple cameras fall dataset.

The later part of this paper is well organized as follows—Section 2 represents the literature discussion on human fall events, Section 3 describes the workflow of the proposed architecture, Section 4 presents experimental details to evaluate the efficiency of the proposed model and Section 5 discusses the limitation of the proposed model along with potential future work.

2. Related Work

In this decade, among different types of deep learning models [10], convolutional neural network (CNN) has acquired immense success in computing like image segmentation [11], object recognition [12], natural language processing [13], image understanding [14] and machine translation [15], which requires a huge training dataset to complete the set up.

A study by Stone, Erik E. and Marjorie S. reveals that falls of older people at home happens most of the time in dark conditions [16]. Sowmya K. and Kang-Hyun Jo [17] classify fall events in a cluttered indoor environment by lessening occlusion effects. However, the knowledge of a series of poses is a key to detect non-fall from fall. For foreground extraction, a frame differencing method is implemented and the human silhouette is extracted using an ellipse fit. Binary support vector machine (SVM) is applied to differentiate fall frames from the non-fall. But SVM underperforms in case of an insufficient training data sample.

Convolutional neural network (CNN) is used to identify different poses in [18]. Here, in different illumination states, the background subtraction method misclassifies some datasets because of shadow. As a result, it generates false predictions in bending, crawling

and sitting positions. Tamura et al. [19] developed a human fall detection system using a gyroscope and an accelerometer. When a fall action is detected it triggers a wearable airbag. To design the system, 16 subjects have been produced to identify mimicked falls and a thresholding technique is applied to perform this action.

A real-life action recognition system is overviewed in [20] using deep bidirectional LSTM (DB-LSTM) and convolutional neural network (CNN). Here, DB-LSTM recognizes hidden sequential patterns in the features where CNN extracts data from video frames. But it shows false projection at the identical background and occluded environment.

Du et al. [21] conducted research on convolutional neural network (CNN) to extract the skeleton joint map from different images. However, the result can be developed if the recurrent neural network (RNN) is implemented properly as in [22]. Anderson et al. [23] procreated a surveillance environment using multiple cameras. Human silhouettes captured from the cameras are converted to 3-D representations known as voxel person. Finally, a fall event is classified from linguistic summarizations of temporal fuzzy inference curves, which represent the states of a voxel person.

Different types of human action can be represented as the movement of the skeleton as the human body is an articulated system. The 2D skeleton is extracted from RGB sequences in [24] using the deeper-cut method and long short term memory (LSTM) is implemented to identify five several actions. However, It is more challenging on processing speed and recognition performance.

The faster R-CNN method is applied in [25], which achieves an accuracy of 95.5% as it cannot properly classify fall events when a person is sitting on a sofa or a chair. A PCANet model is trained followed by SVM classifier in [26], which obtains less sensitivity of 88.87% as it cannot identify fall events properly. Moreover, SVM underperforms as fine-tuning hyperparameters in SVM is not so easy.

In [27] curvature scale space (CSS) features are extracted from human silhouettes and an extreme learning machine (ELM) classifier is used to identify fall action. However, this experiment achieves 86.83% accuracy as it misclassifies the lying and walking position of humans as the silhouette of a falling and a lying person is similar.

A two-stage human fall classification model is proposed in [28], which achieves an accuracy of 97.34%. To identify human posture from the human skeleton, at the preprocessing stage it considers deflection angles along with spine ratio. To extract the human skeleton it uses OpenPose and to identify confusing daily living actions from fall events, a time-continuous recognition algorithm is developed. However, this model misclassifies workout motions and for increasing accuracy, it has proposed to develop a deep learning model in the future.

Besides, Chen et al. [29] detect human fall events from the human skeleton information by OpenPose and obtains an accuracy of 97%. To identify fall action three efficient parameters are considered here like centerline angle between human body and ground, speed of descent at the center point of the hip joint, the ratio of width and height of external rectangle surrounding the human body. However, this model is unable to classify partially occluded human actions.

Moreover, A vision-based human fall detection model is proposed by Chen et al. [7] in the case of complex background. They perform the mask R-CNN method to extract the object from a noisy background. Afterward, for fall action detection, attention-guided Bi-directional LSTM is applied and it acquires 96.7% accuracy. However, this model cannot identify the behavior of multiple people living in the same room.

Nowadays, different types of wearable sensors, that is, accelerometers, buttons are most frequently used to detect falls at the right time. However, using such detectors is uncomfortable and most of the time, older people forget to wear these sensors. Moreover, using a help button is worthless if the person has fainted or is immobilized. Such a framework for elderly fall classification and notification is proposed in [9]. Here, the tri-axial acceleration of human movement is measured with a cell phone. Both the time domain and frequency domain features are considered here. After feature extraction

and pre-processing they have performed a deep belief network with a view to training and testing the system. It shows an accuracy of 97.56% sensitivity along with 97.03% specificity. However, for elder people, it is not always possible to carry a cell phone in an indoor environment.

In [8], the authors conducted a highly promising experiment to develop a hybrid model using a machine learning method combining with deep learning. After analyzing different comparative experiments, they proposed an architecture of CNN with LSTM for posture detection with an accuracy of 98%. This CNN model has been designed with 10 layers without any batch normalization. Unnormalized data and less dropout rate of 20% can lead to a huge training time of the dataset and the performance of the model may also be affected.

Table 1 represents the summarization of this literature discussion.

Table 1. Summarization of literature discussion.

Research Paper	Proposed Models	Limitations	Accuracy (%)
[17]	Frame differencing for foreground extraction, ellipse fit for human silhouettes extraction and SVM classifier for fall classification.	SVM underperforms in case of an insufficient training data sample.	96.34%
[18]	Background subtraction method to extract foreground and CNN to classify fall events.	Generates false predictions in bending, crawling, and sitting positions.	90.2%
[20]	CNN for feature extraction and DB-LSTM recognizes sequential pattern.	Shows false projection at the identical background and occluded environment.	92.66%
[21]	CNN to extract the skeleton joint map.	The result can be developed if the recurrent neural network is incorporated.	94%
[23]	Human silhouettes are converted to voxel person and linguistic summarizations of temporal fuzzy inference curves classify fall events.	Not applicable for short-term activity recognition.	96.5%
[24]	Deeper-cut method extracts 2D skeleton and LSTM identify fall actions.	Low accuracy rate.	90%
[25]	Faster R-CNN for fall classification.	Can not properly classify fall events when a person sitting on a sofa or a chair.	95.5%
[26]	PCANet model is trained followed by SVM classifier.	Low accuracy rate.	88.87% (Sensitivity)
[27]	CSS features are extracted from human silhouettes and extreme learning machine (ELM) classify fall events.	Misclassifies the lying position.	86.83%
[28]	OpenPose method extract human skeleton and time-continuous recognition algorithm identify fall event.	Misclassifies workout motions.	97.34%
[29]	OpenPose identify fall events using information from human skeleton.	Unable to classify partially occluded human actions.	97%
[7]	Mask R-CNN extract object from noisy background and Bi-LSTM classify human actions.	Cannot identify the behavior of multiple people living in the same room.	96.7%
[9]	Deep belief network for training and testing human actions.	Not always possible to carry a cell phone in an indoor environment.	97.56% (Sensitivity)
[8]	CNN integrated with LSTM for posture detection.	Unnormalized data may lead to huge training time.	98%

As deep learning models have outperformed state-of-the-art models, there are many scopes for innovation and development in this research area. However, a computer vision-based model is proposed to classify fall events at the right time, which provides whole information regarding the movement of a person.

3. Workflow of Proposed Architecture

Elderly people monitoring is done through a digital video camera, which will be placed in the room as there are some distance limitations in the Kinect camera. Although it has some privacy concerns, it will give us information about surroundings in case of fall events. Sequential frames are generated from videos of variable length. Frames are passed into the convolutional neural network to extract key features. After that these features are passed into a gated recurrent unit. The output from GRU is passed to a sigmoid classifier to predict the class. Figure 1 illustrates an overview of the proposed network.

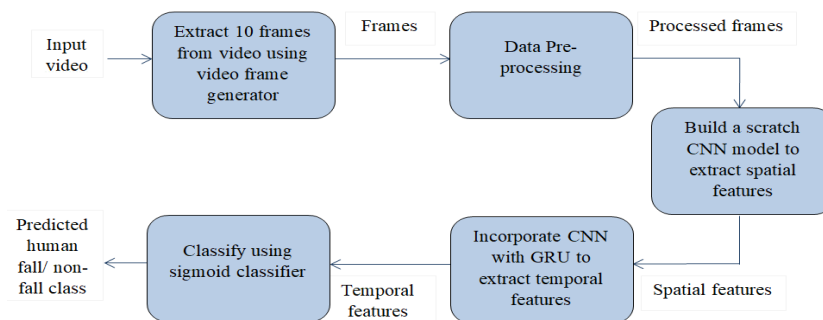


Figure 1. Workflow of the proposed human fall classification model.

3.1. Frame Generation from Video

There are approximately 300 videos from different datasets of various durations. From each video, we need to pass a sequence of frames to CNN. As mentioned earlier, we picked 10 distributed images from the whole video rather than considering all other frames. The algorithm to generate frame sequence from video is given below. We performed a frame differencing method in Algorithm 1 along with other operations for tweaking important and informative frames from the video.

Algorithm 1 Video to frame generation.

```

Input: Number of frames need to extract
Output: Batches of images
move_detect = Statistical mean threshold value
If move_detect > 0
f = frames [0]
f = convert image from RGB to GRAY
last = f
important_frame = []
for frame number i = 1 to length(frames)
f = frame[i]
cp = convert image from RGB to GRAY
delta = absolute difference(cp, last)
threshold = threshold ( delta, thresholdvalue, 255, threshold_binary)
threshold = dialate (threshold, structure element, iteration)
if np.mean(threshold) > move_detect
mark as important_frame
end
last = cp
end
nb_ignore = length (important_frame)/nb_frame
if length(important_frame) > nb_frame
collect frame from last and ignore nb_ignore
end
end
  
```

3.2. Preprocessing

For improving image properties, eliminating noisy artifacts and enhancing certain features, it is necessary to preprocess data. We have performed preprocessing by three steps—resizing, augmentation and normalization. In order to minimize computational cost frames are resized to 150×150 .

On the resized image, augmentation is performed, which transforms frames at each epoch of training. For augmentation, we have performed zoom, horizontal flip, rotation, width shift and height shift. This helps for better generalization of this model.

3.3. Convolutional Neural Network

Convolutional neural network is a type of deep neural network which extracts key features from images using learnable weights and biases and can differentiate one object from another. Comparing with other classification algorithms, it requires much lower preprocessing of images. It consists of an input layer, an output layer along with a series of hidden layers. The hidden layers typically consist of a stack of convolutional layers which perform pixel-wise convolution or multiplication operation and generate a convolved image. The activation function used here is a rectified linear unit (ReLU) layer which is followed by a series of pooling layers and fully connected layers.

There are three types of the pooling operation. These are max pooling, min pooling, and average pooling operation. Among these, the max-pooling operation is most frequently used as it minimizes computational cost along with learnable parameters.

For extracting features from images, CNN uses this series of convolutional layers followed by different pooling layers, flatten layer as well as fully connected layer. Each layer works with different activation functions. Following this, the proposed architecture is also designed with a series of convolution layers with 'ReLU' activation function. The ReLU activation function generates a rectified feature map as output. It does not excite all neurons at a time. When the output of linear transformation becomes zero, the neurons will be discarded. Although there are different types of pooling layers, we have used the max-pooling layer

3.4. Custom 2DCNN-GRU Architecture

Preprocessed data are fed to CNN. We have proposed a CNN architecture for extracting spatial features. As the environment in the dataset is less complex, 16, 32, 64, 128, 256, and 512 kernels are used sequentially in the convolution layer to extract features from the image. Weights are tuned for each layer depending on the activation function. This network gives much more accuracy than other pre-trained networks of CNN because an experiment is done for choosing kernel initializer along with activation function. In this model he_uniform kernel initializer is used for choosing the initial kernel and updating weights for training data with ReLU activation function. We have also conducted an experiment for choosing appropriate momentum for batch normalization. We have performed batch normalization with a momentum of 0.9. Batch normalization standardizes the mean and variance to make learning stable. Here, it takes 83 s to complete an epoch using batch normalization. Without batch normalization, it takes 99 s for an epoch in case of training the dataset. After batch normalization, the max pooling operation with stride (2, 2) is performed to extract the strong edges in the image. This also helps to remove shallow edges corresponding to noise. The output of the convolution layer and max-pooling layer are shown, respectively, in Figures 2 and 3. Figure 2 depicts the convolved image and Figure 3 illustrates the pooled feature map.

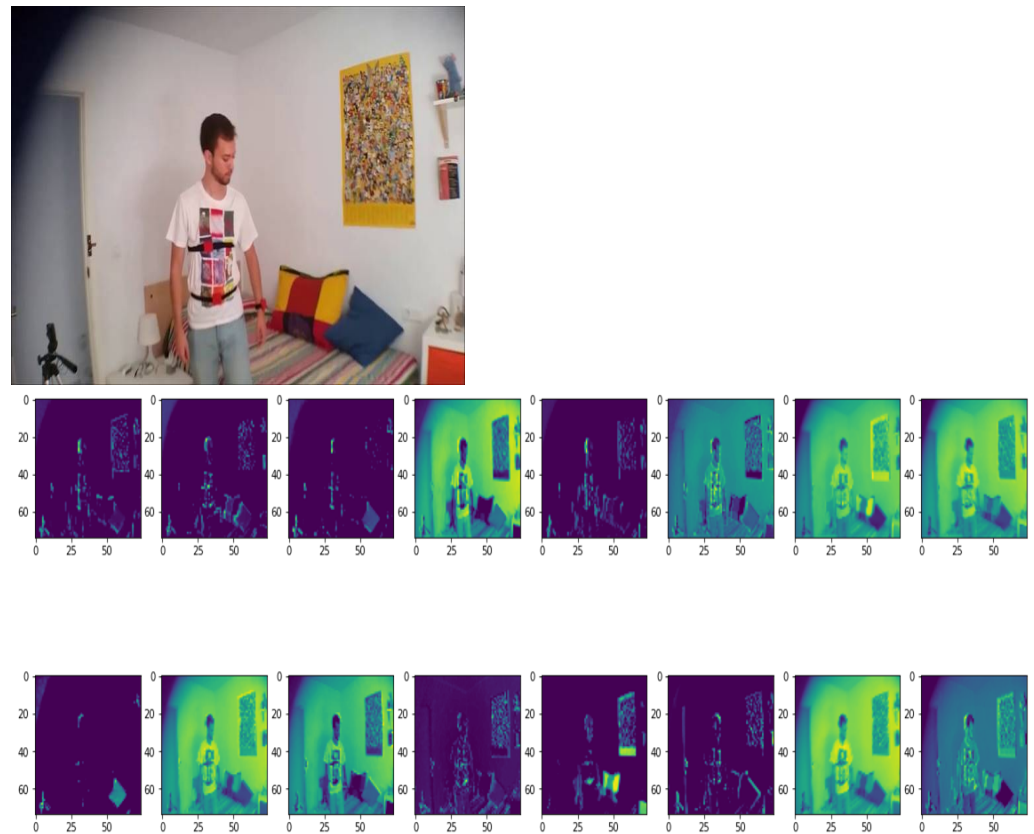


Figure 2. Visualization of the output of convolution layer for human fall classification.

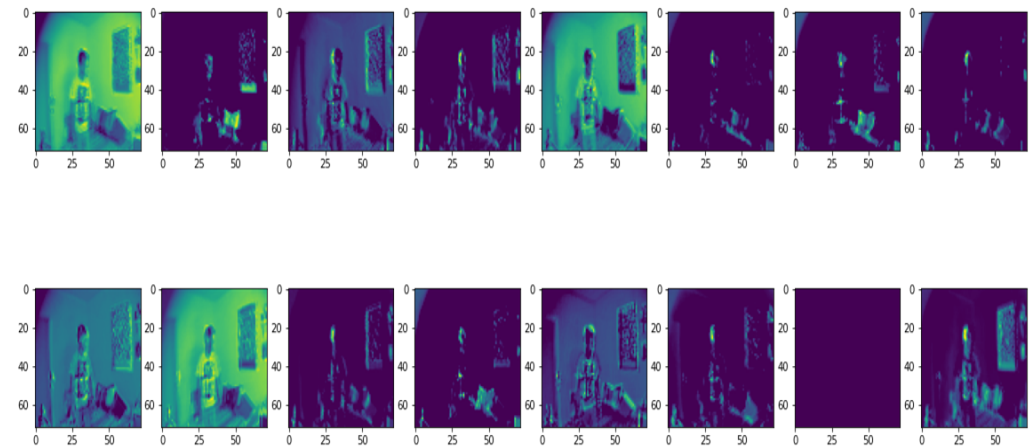


Figure 3. Visualization of the output of max-pooling layer for human fall classification.

After that, a time-distributed layer is used with 512 nodes to prepare data for RNN. Then GRU cells are used to follow the temporal dependency of video frames. The output of the GRU cell is passed to the dense layer. For eliminating overfitting, we have experimented on the dropout rate which is described in this paper. In this regard, a dropout rate of 0.5 is used in the dense layer for better accuracy. Adam optimizer with a learning rate of 0.0001 is used in the model. The detail of the proposed 2DCNN-GRU architecture is depicted in Figure 4.

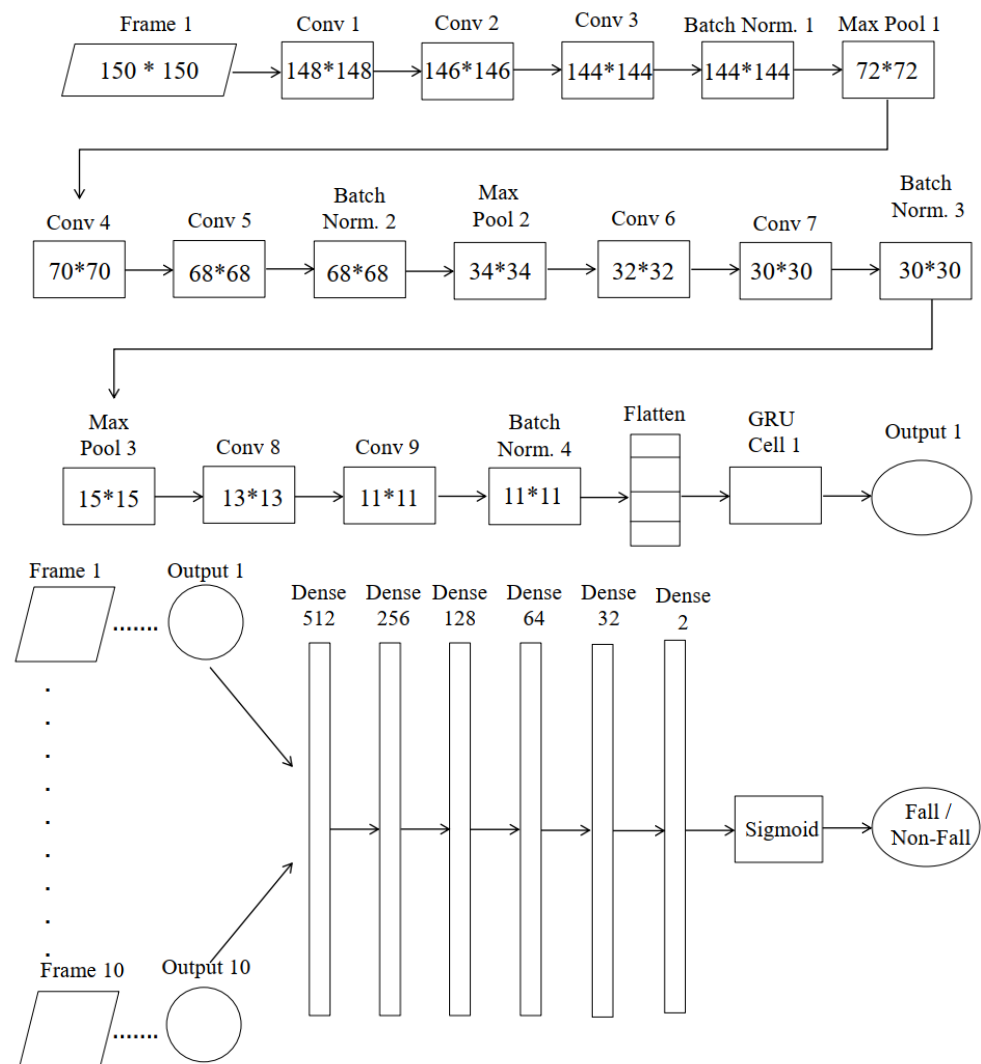


Figure 4. Proposed 2DCNN-GRU architecture to classify human fall.

3.5. Gated Recurrent Unit (GRU)

Among all kinds of human movements, to classify fall events, we need to consider temporal features along with spatial features. The recurrent neural network can extract temporal features by remembering necessary information from the past. However, during this operation, it faces vanishing and exploding gradient problems. In our model, we have used the gated recurrent unit (GRU) network to solve this problem of RNN using an update gate and a reset gate, which are the vectors to decide what information needs to be passed as output. These gates can be trained to hold information from long ago, without erasing new input through time but pass relevant information for prediction to the next time steps. GRU gives a much better result than a long short term memory (LSTM) cell because of its simple architecture.

Figure 5 depicts the architecture of a GRU cell. There are three gates in GRU called the Update gate, the Reset gate, and the Current memory gate, which has no internal cell state. The Update gate identifies significant preceding information from the antecedent time steps, which is required to be passed along with future information. The reset gate decides how much of the past information needs to be forgotten. The current state gate determines the current state information along with the relevant information from the past.

Mathematical equations learned from [30] are given below which are used to perform these operations:

$$\text{Update gate, } Z_t = \sigma(W^{(Z)}x_t + U^{(z)}h_{(t-1)}) \quad (1)$$

$$\text{Reset gate, } r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{(t-1)}) \tag{2}$$

$$\text{Current memory gate, } \hat{h}_t = \tanh(Wx_t + r_t * Uh_{(t-1)}) \tag{3}$$

$$\text{Final memory, } h_t = Z_t * h_{(t-1)} + ((1 - Z_t) * \hat{h}_t). \tag{4}$$

Here, x_t represents the mini batch input, $h_{(t-1)}$ acts as the hidden state of utmost time step, $W^{(z)}$ and $W^{(r)}$ are current weight parameters and $U^{(z)}$, $U^{(r)}$ are updated weight parameters.

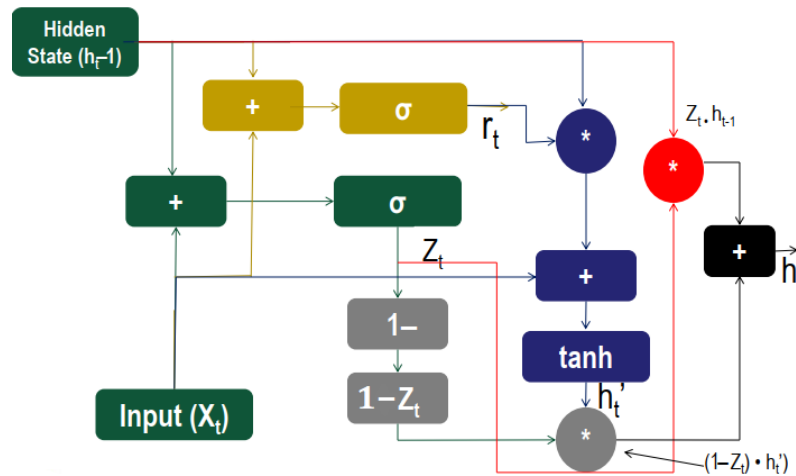


Figure 5. Gated Recurrent Unit (GRU) Cell.

The output of GRU cell is passed to the dense layer with a dropout rate of 50%. Finally, sigmoid activation function [31] is applied for binary classification of fall and non-fall action through the following equations.

$$\text{Sigmoid activation function, } S(x) = e^x / (e^x + 1). \tag{5}$$

Deep learning models are efficiently used to lessen uncertainty and entropy measures the uncertainty level. Binary cross entropy plays a significant role to alleviates incongruity between the explorative distribution of training data and the distribution incited by the model. Here following binary cross-entropy loss function [32] equation is used to compare the predicted class with the actual class.

$$\text{Binary cross entropy loss, } L = - \sum_{i=1}^2 t_i \log(p_i). \tag{6}$$

Here, t_i denotes the truth value and p_i represents the sigmoid probability of i th class.

4. Experiments

For executing this experiment, we have used the machine of a configuration of AMD Ryzen 7 2700X Eight-core 3.7 GHz Processor, 32 GB RAM, NVIDIA GEFORCE RTX 2060 SUPER of 8 GB GPU memory. We have also used Google colab for faster execution.

4.1. Dataset Description

We have conducted this experiment on the UR fall detection dataset and multiple cameras fall dataset to examine the accuracy of our proposed model.

4.1.1. UR Fall Detection Dataset

UR fall detection dataset is one of the most benchmark datasets which comprises 70 indoor videos among which 30 fall events and 40 daily living activities. Here, fall

activities are recorded using two Kinect cameras whereas daily living activities are recorded using only one camera. In this dataset, there are 30 frames per video. Each video possesses a resolution of 640×240 .

4.1.2. Multiple Cameras Fall Dataset

Multiple cameras fall dataset is one of the most extensive datasets which is widely used to classify human fall action. It includes 192 videos where 96 videos represent fall events and 96 videos are of regular indoor activities. This dataset is recorded with 24 scenarios which represent 9 different activities like walking, falling, lying on the ground, crouching, and so on. Eight different cameras are used to capture each activity. Each video has a frame rate of 30 fps with a frame size of 720×480 .

4.2. Evaluation Metrics

The proposed model is also evaluated in terms of accuracy, precision, sensitivity, specificity and F1-score. Accuracy demonstrate the rate of correctly classified data using following equation [33]:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN), \quad (7)$$

where TP represents the true positive rate, that is, the detected result is a fall, which is actually a fall event, TN represents the true negative rate, that is, the detected event is a non-fall, which is actually a non-fall event, FP represents the false positive rate, that is, the detected event is a fall, which is a non-fall event in real-time and FN stands for the false negative, which means that the detected result is a non-fall but it is actually a fall event of the human being. In the case of binary classification, 100% precision score signifies that every element of the positive class verily belongs to the positive class, which is calculated by the equation [33]:

$$Precision = TP / (TP + FP). \quad (8)$$

Sensitivity gives the probability of positive result of test data through the equation [33]:

$$Sensitivity = TP / (TP + FN). \quad (9)$$

The equation [33] to calculate specificity is given below which provides the probability of negative result of test data.

$$Specificity = TN / (TN + FP). \quad (10)$$

F1-score implies the harmonic mean between precision and sensitivity. The following equation, Ref. [33] is used to calculate the F1-score of the test data.

$$F1 - score = (2 * Precision * Sensitivity) / (Precision + Sensitivity). \quad (11)$$

4.3. Results and Discussion

To implement this scratch model, we have made several experiments on the percentage of training and validation datasets to achieve better accuracy and we have got a mean accuracy of 99% where 99.8% and 98% accuracy for the UR fall detection dataset and Multiple cameras fall dataset, respectively. Multiple cameras fall dataset give less accuracy than the UR fall detection dataset because of the large number of videos in multiple cameras fall dataset. In multiple cameras fall dataset, our model misinterprets when a person lies down intentionally. Our proposed model outperforms when validation data is below 50%, which is shown in Figures 6–8. Considering this, 35% videos are used for validation and 25% are used for testing. The rest of the data are used for training.

Figures 6–8 illustrate that at 40th epoch, the model achieves the best training and validation accuracy for 40% training, 35% validation and 25% testing data.

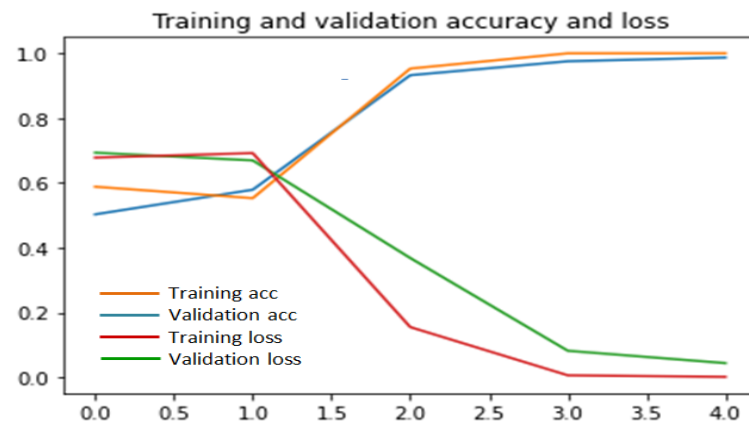


Figure 6. Performance of scratch model for 35% validation and 25% test data.

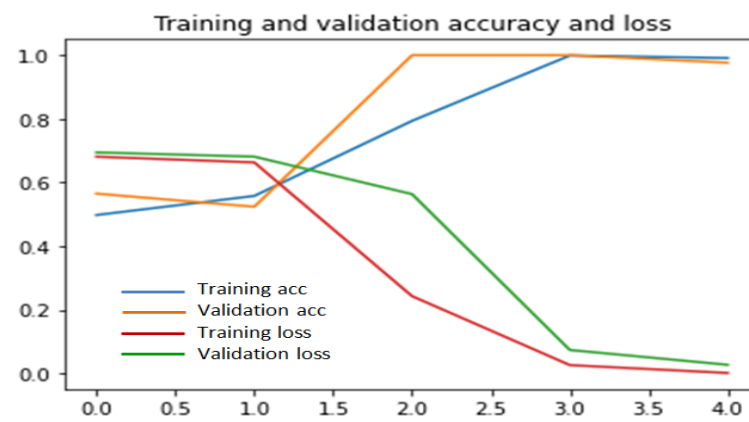


Figure 7. Performance of scratch model for 50% validation and 20% test data.

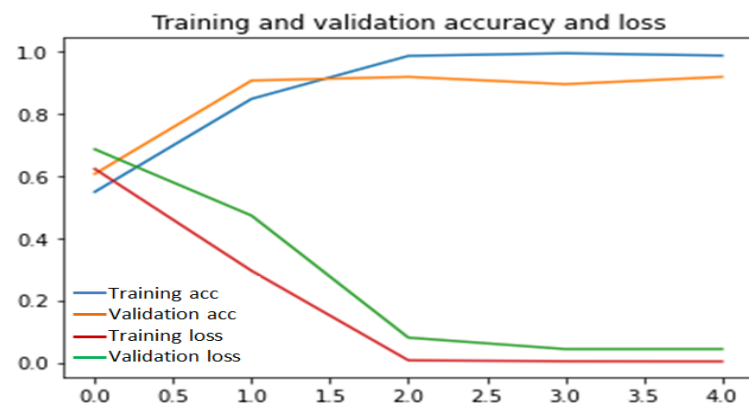


Figure 8. Performance of scratch model for 60% validation and 20% test data.

An experiment is done using different frame numbers illustrated in Figure 9. This reveals that fewer than eight frames per video decrease the accuracy rate because it cannot consider all significant frames. Similarly, more than 18 frames per video decrease processing speed because of the large number of computations. The changes of the execution time of the proposed model according to the number of frames in input can be described using Table 2. Here, we see that the fewer the number of frames in input, the faster the execution of the model. However, it decreases the accuracy rate because of the absence of enough information according to Figure 9.

Therefore we have chosen 10 numbers of distributed frames from the whole video to reduce computational complexity along with better accuracy throughout the whole

experiment. Figures 10 and 11 illustrate this 10 numbers of frame sequences of an input video of daily activities and fall events respectively in multiple cameras fall dataset.

Table 2. Effect of number of frames on execution time.

Number of Frames	Execution Time of Proposed Model
5	2.8 min
8	3.9 min
10	4.7 min
12	5.4 min
15	6.8 min
18	7.8 min
20	9.4 min
22	11.7 min

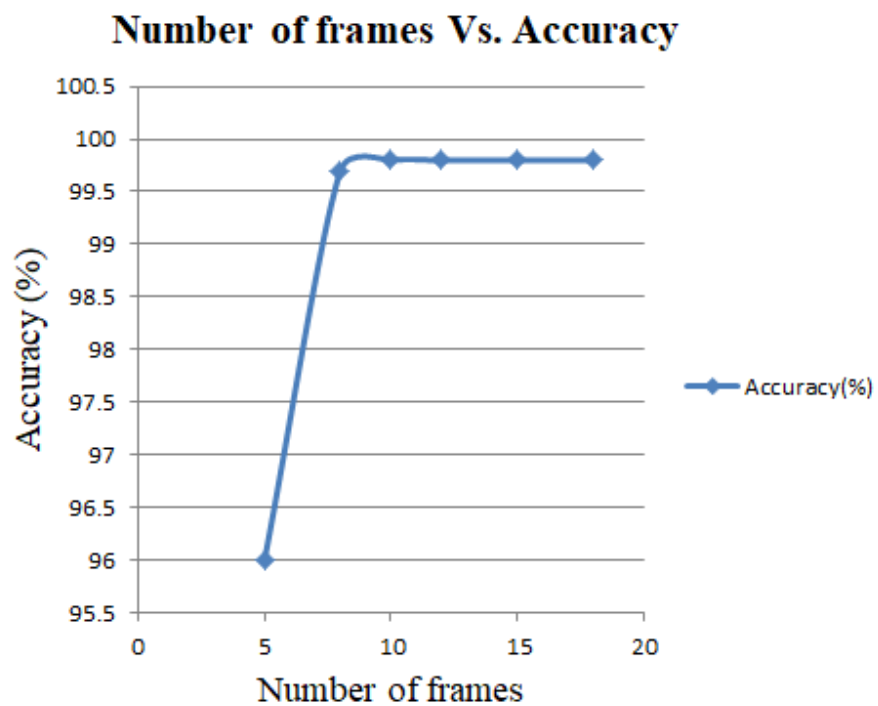


Figure 9. Number of frames vs. validation accuracy curve.



Figure 10. Sequential video frames for daily activity in multiple cameras fall dataset.

As stated earlier, in this scratch model we have performed batch normalization to achieve faster convergence during training. Therefore, Table 3 evaluates the performance

using normalized and unnormalized data. It depicts that normalized data achieve a high accuracy rate within a lower number of epochs than unnormalized data.

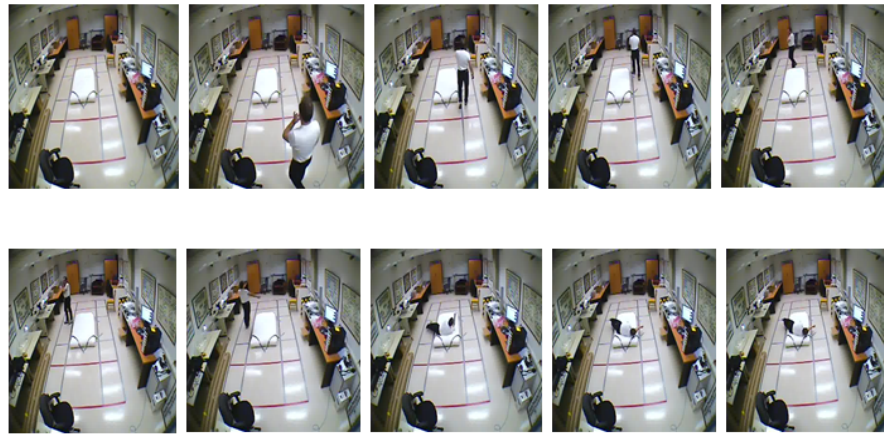


Figure 11. Sequential video frames for fall event in multiple cameras fall dataset.

Table 3. Effects of batch normalization for performance measurement.

	Training Accuracy	Validation Accuracy	Test Accuracy	Total Epochs
Normalized data	100%	99.7%	99%	40
Unnormalized data	94%	84%	81%	55

Output of different layers in 5 distributed frames among 10 frames are illustrated in Figure 12. Low to high level feature maps in Figure 12 shows that a deep neural network acts like a black-box, which extracts features of the input image.

At the preliminary stage, the layers extract shallow features like colors and lines while the deep level layers extract the more details patterns. Higher level features are encoded that are difficult to extract and are presented to human readable format [34]. Therefore, deep learning models can extract higher level features that are mystical and more immense in amount than the features considered by humans.

Figure 13 shows the output of this experiment to classify human fall events. Here, the first 4 rows represent 5 frames of 4 non-fall event's videos and the rest of the rows represent 5 frames of 4 fall event's videos. Figure 14 demonstrates the attention map where the blue region depicts the region of interest after executing this model, which classifies fall and non-fall events.

Here, a dropout rate of 50% is used in dense layers because more or less than 50% dropout rate gives lower accuracy along with the overfitting problems of training data. The impact of the dropout rate for the change in accuracy is illustrated in Figure 15. This figure illustrates that the proposed model achieves a maximum accuracy of 99.8% at the dropout rate of 50% for the UR fall detection dataset.

Tables 4 and 5 represent a summary of test accuracy for different existing models where VGG 16 and VGG 19 give 98% test accuracy, Xception generates 99% accuracy but it uses a huge number of parameters compared with our proposed model. As VGG 16, VGG 19, and Xception are pre-trained models, so the number of parameters and depth of these models are observed from [35] where the rest of the others are known experimentally. In [36], the authors classify human fall action using 3DCNN combining with LSTM and obtains an accuracy of 99%. However, 3DCNN comprised with LSTM needs a huge number of parameters. Moreover, training iteration for 2DCNN needs 0.5 s per pass where 3DCNN lasts for 3 s. We have also conducted an experiment using 2DCNN with LSTM and it gives an accuracy of 89% for the same dataset.

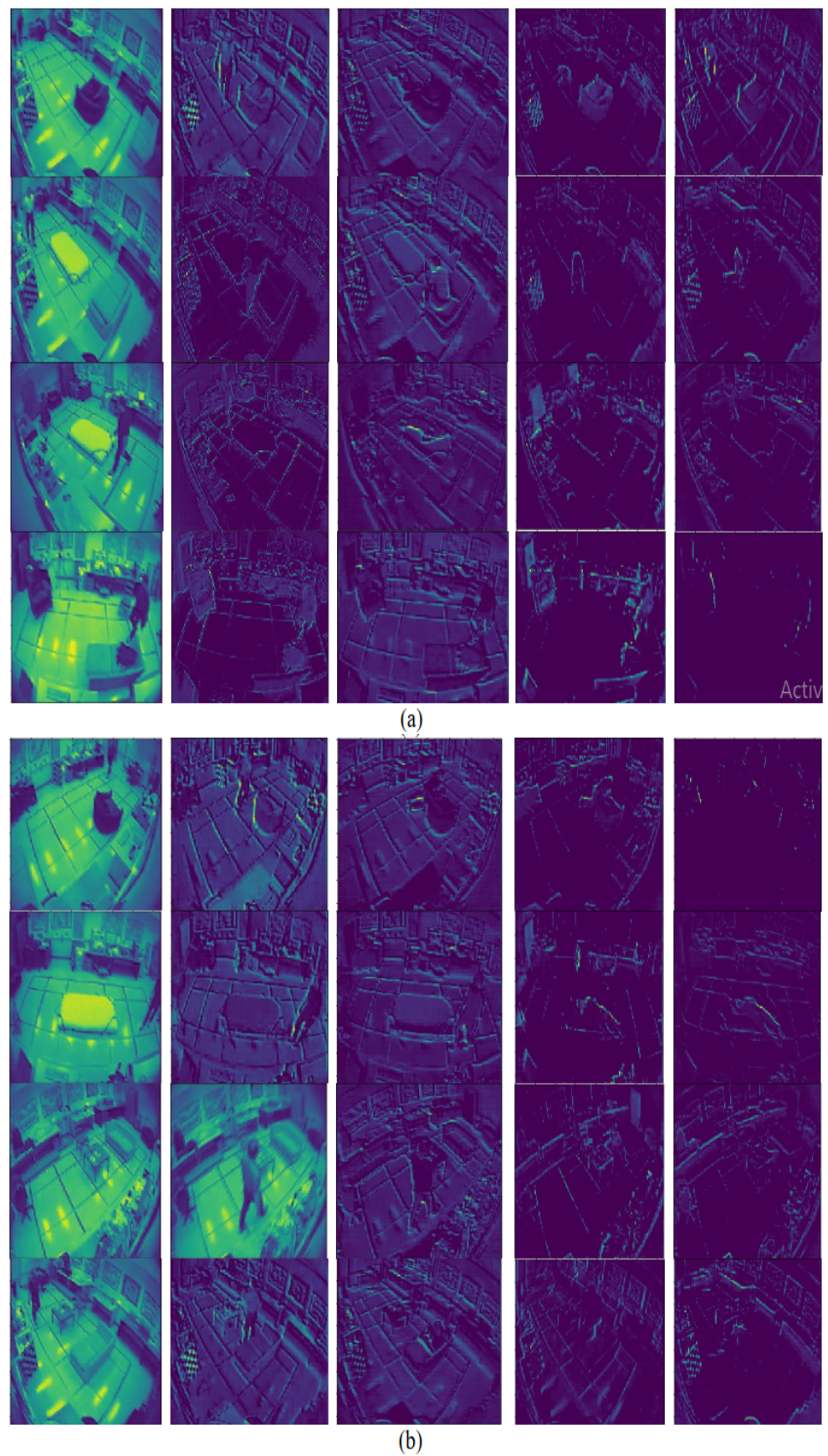


Figure 12. Each row showing low-to-high level feature maps using the proposed convolutional neural network (CNN) model in multiple cameras fall dataset for (a) daily activities, (b) fall events.

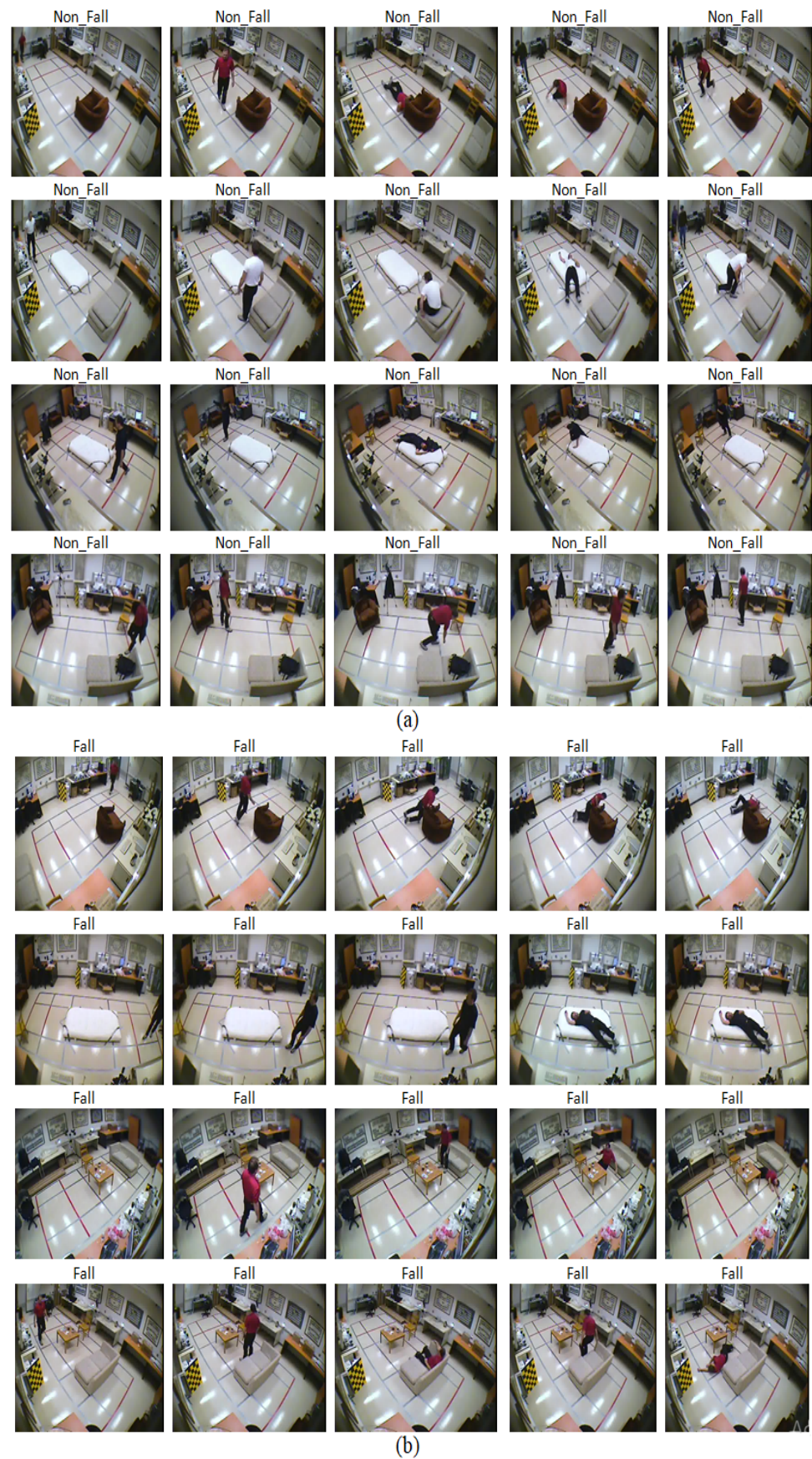


Figure 13. Output of the proposed model for multiple cameras fall dataset for (a) non-fall events, (b) fall events.

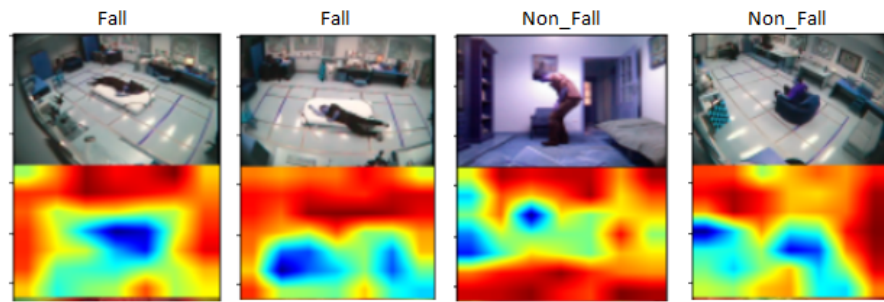


Figure 14. Attention map for fall classification.

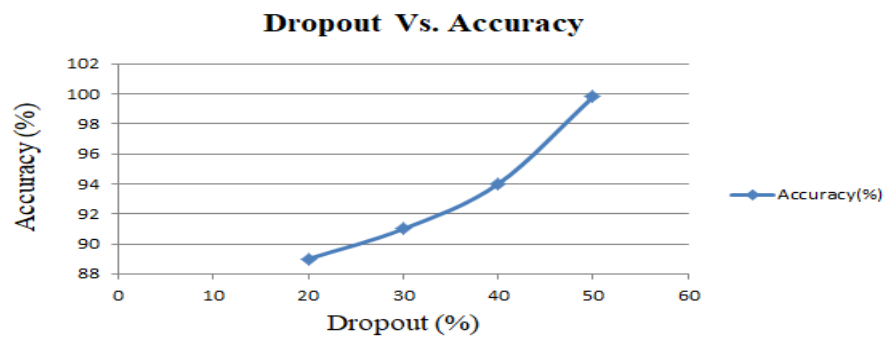


Figure 15. Dropout vs. validation accuracy curve.

Table 4. Comparison of test accuracy of the proposed model with existing models in the UR fall detection dataset for human fall classification.

Existing Models	Accuracy (%)
VGG 16	98%
VGG 19	98%
Xception	99%
1DCNN with GRU	94.30%
2DCNN with Bi-LSTM	95.50%
3DCNN with LSTM	99%
2DCNN with LSTM	89%
Proposed Model	99.80%

Table 5. Comparison of test accuracy of the proposed model with existing models in multiple cameras fall dataset for human fall classification.

Existing Models	Accuracy (%)
VGG 16	97.60%
VGG 19	98%
Xception	98%
1DCNN with GRU	92.70%
2DCNN with Bi-LSTM	95%
3DCNN with LSTM	97.50%
2DCNN with LSTM	88%
Proposed Model	98%

It is difficult to estimate the training time of parameters of models as it depends on the GPU model. Using our hardware configuration mentioned earlier and GPU, we have executed some deep learning approaches. Number of parameters, depth, and the training times for these models are represented in Table 6. From this table, it is seen that the training time of our proposed model is faster because of its fewer parameters than others.

Table 6. Number of parameters, depth and training time comparison with existing models for human fall classification.

Existing Models	Number of Parameters	Depth	Training Time
VGG 16	138,357,544	23	39 m 21 s
VGG 19	143,667,240	26	46 m 31 s
Xception	22,910,480	126	18 m 22 s
3DCNN with LSTM	12,317,230	20	11 m 44 s
2DCNN with LSTM	7,523,320	18	7 m 16 s
Proposed Model	5,288,860	18	4 m 7 s

Considering all these issues, our proposed model gives an average of 99% accuracy using a lower number of parameters along with depth. This is illustrated by the confusion matrix in Figures 16 and 17 for the different datasets. The elements in dark blue diagonal demonstrate the number of correctly classified Fall and Non-Fall events. Here, we see the global accuracy is 100%, which is the ratio of the summation of elements in diagonal by the summation of all elements in the entire matrix. From the confusion matrix for UR fall detection dataset in Figure 16, we see that there are 25 test data points and all are classified correctly. In the confusion matrix for multiple cameras fall dataset in Figure 17, there are 48 test data points. Among them, one non-fall video is misclassified as a fall event.

True Label	Fall	15	0
	Non-Fall	0	10
		Fall	Non-Fall
		Predicted Label	

Figure 16. Confusion matrix for UR fall detection dataset.

True Label	Fall	22	0
	Non-Fall	1	25
		Fall	Non-Fall
		Predicted Label	

Figure 17. Confusion matrix for Multiple cameras fall dataset.

The class-wise performance like accuracy rate, precision score, sensitivity, specificity, and F1-score of our proposed 2DCNN-GRU model using the UR fall detection dataset and the multiple cameras fall dataset is shown in Tables 7 and 8.

Table 7. Class-wise performance of the scratch model for UR fall detection dataset.

Classes	Mean Accuracy (%)	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Fall event	100	100	100	100	100	100
Non-fall event		100	100	100	100	100

Table 8. Class-wise performance of the scratch model for Multiple cameras fall dataset.

Classes	Mean Accuracy (%)	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Fall event	98	100	100	96	100	98
Non-fall event		96.15	100	96	100	98

Tables 9 and 10 illustrate the comparison of the performance of our proposed model with some existing models. Using the UR fall detection dataset, our model achieves a maximum accuracy of 99.8% compared with Kasturi et al. [17] and Lu et al. [36], which are based on support vector machine (SVM) and 3DCNN followed by LSTM, respectively. On the multiple cameras fall dataset, the proposed model outperforms, acquiring 98% accuracy compared to Wang et al. [26], which is based on PCANet and it is also much higher than Ma et al. [27] based on extreme learning machine (ELM).

Table 9. Performance comparison with existing models using UR fall detection dataset.

Methods	Accuracy (%)
Kasturi et al. [17]	96.34%
Lu et al. [36]	99.27%
Proposed model	99.8%

Table 10. Performance comparison with existing models using multiple cameras fall dataset.

Methods	Accuracy (%)
Wang et al. [26]	96%
Ma et al. [27]	97.2%
Proposed model	98%

5. Conclusions

As the deep learning algorithm outperforms other feature extraction algorithms, in this paper, we have proposed a combined architecture where CNN is incorporated with GRU. This is quite challenging to identify human falls at the right time, not only to minimize the negative consequences of a fall but also to increase acceptance level among elderly people. We have used two existing benchmark datasets of fall classification—UR fall detection dataset and multiple cameras fall dataset of variable length videos. The frame number from each video is chosen empirically. To perform classification on these datasets, a scratch model is proposed where CNN followed by GRU is performed, which is our key contribution. Another novelty of our work is we have conducted some transfer learning models along with other deep learning models using the same datasets. However, the proposed model outperforms with an accuracy of 99% with a lower number of parameters than other existing architecture by parameter tunings. Binary cross-entropy loss function outperforms others like mean squared error, hinge loss, and so forth. Despite these, this experiment would be much better if we could enrich our dataset. In the future, we can

include an alarm system to take necessary action in time and reduce the after-fall effects. Moreover, the proposed architecture extracts features from the entire frame. Besides, we can reduce the computation time if we can consider only the salient region. Furthermore, this model can also collaborate with people counting techniques to identify the actions of different people residing in the same room.

Author Contributions: All authors contributed equally to the conception of the idea, the design of experiments, the analysis and interpretation of results, and the writing and improvement of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The authors have used publicly archived dataset named UR Fall Detection Dataset and Multiple cameras fall dataset for validating the experiment. The dataset is available at <http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html> (accessed on 18 February 2021) and <http://www.iro.umontreal.ca/~labimage/Dataset/> (accessed on 18 February 2021).

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
ReLU	Rectified Linear Unit
DB-LSTM	Deep Bi-directional Long Short Term Memory

References

- Mubashir, M.; Shao, L.; Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152. [[CrossRef](#)]
- Yang, L.; Ren, Y.; Hu, H.; Tian, B. New fast fall detection method based on spatio-temporal context tracking of head by using depth images. *Sensors* **2015**, *15*, 23004–23019. [[CrossRef](#)]
- Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 611–622. [[CrossRef](#)]
- Lord, S.; Smith, S.; Menant, J. Vision and falls in older people: Risk factors and intervention strategies. *Clin. Geriatr. Med.* **2010**, *26*, 569–581. [[CrossRef](#)]
- Canada's Aging Population. Public Health Agency of Canada, Division of Aging and Seniors. 2002. Available online: <http://www.publications.gc.ca/site/eng/9.648495/publication.html> (accessed on 26 February 2021).
- Reports on Senior's Falls in Canada. Public Health Agency of Canada, Division of Aging and Seniors. 2005. Available online: <https://www.canada.ca/en/public-health/services/health-promotion/aging-seniors/publications/publications-general-public/seniors-falls-canada-second-report.html> (accessed on 26 February 2021).
- Chen, Y.; Li, W.; Wang, L.; Hu, J.; Ye, M. Vision-Based Fall Event Detection in Complex Background Using Attention Guided Bi-Directional LSTM. *IEEE Access* **2020**, *8*, 161337–161348. [[CrossRef](#)]
- Liaqat, S.; Dashtipour, K.; Arshad, K.; Assaleh, K. A Hybrid Posture Detection Framework: Integrating Machine Learning and Deep Neural Networks. *IEEE Sens. J.* **2021**, *21*, 9515–9522. [[CrossRef](#)]
- Jahanjoo, A.; Naderan, M.; Rashti, M.J. Detection and Multi-class Classification of Falling in Elderly People by Deep Belief Network Algorithms. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 4145–4165. [[CrossRef](#)]
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
- Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [[CrossRef](#)]
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 770–778.
- Sarikaya, R.; Hinton, G.E.; Deoras, A. Application of deep belief networks for natural language understanding. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 778–784. [[CrossRef](#)]
- Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [[CrossRef](#)] [[PubMed](#)]

15. Geoffrey, H.; Li, D.; Dong, Y.; George, D.; Abdel-rahman, M.; Navdeep, J.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97.
16. Stone, E.E.; Marjorie, S. Fall Detection in Homes of Older Adults Using The Microsoft Kinect. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 290–301. [[CrossRef](#)]
17. Kasturi, S.; Jo, K.H. Classification of Human Fall In Top Viewed Kinect Depth Images Using Binary Support Vector Machine. In Proceedings of the 10th International Conference on Human System Interactions (HSI), Ulsan, Korea, 17–19 July 2017; pp. 144–147.
18. Alhimale, L.; Zedan, H.; Al-Bayatti, A. The Implementation of An Intelligent and Video-Based Fall Detection System Using a Neural Network. *Appl. Soft Comput.* **2014**, *18*, 59–69. [[CrossRef](#)]
19. Tamura, T.; Yoshimura, T.; Sekine, M.; Uchida, M.; Tanaka, O. A wearable airbag to prevent fall injuries. *IEEE Trans. Inf. Technol. Biomed.* **2009**, *13*, 910–914. [[CrossRef](#)] [[PubMed](#)]
20. Ullah, A.; Ahmad, J.; Muhammad, K.; Baik, S.W. Action Recognition in Video Sequence Using Deep Bi-Directional LSTM With CNN Features. *Vis. Surveill. Bioinform.* **2018**, *6*, 1155–1165. [[CrossRef](#)]
21. Du, Y.; Fu, Y.; Wang, L. Skeleton Based Action Recognition with Convolutional Neural Network. In Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition, Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 579–583.
22. Insafutdinov, E.; Pishchulin, L.; Andres, B.; Andriluka, M. DeeperCut: A Deeper, Stronger and Faster Multi-Person Pose Estimation Model. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 34–50.
23. Anderson, D.; Luke, R.; Keller, J.; Skubic, M.; Rantz, M.; Aud, M. DeeperCut: Linguistic Summarization of Video for Fall Detection Using Voxel Person And Fuzzy Logic. *Comput. Vis Image Underst* **2009**, *113.1*, 80–89. [[CrossRef](#)]
24. Lie, W.N.; Le, A.T.; Lin, G.H. Human Fall Down Event Detection Based on 2D Skeletons and Deep Learning Approach. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; pp. 1–4.
25. Min, W.; Cui, H.; Rao, H. Detection of Human Falls on Furniture Using Scene Analysis Based on Deep Learning and Activity Characteristics. *IEEE Access* **2018**, *6*, 9324–9335. [[CrossRef](#)]
26. Wang, S.; Chen, L.; Zhou, Z.; Sun, X.; Dong, J. Human fall detection in surveillance video based on PCANet. *Multimed. Tools Appl.* **2016**, *75*, 11603–11613. [[CrossRef](#)]
27. Ma, X.; Wang, H.; Xue, B.; Zhou, M.; Ji, B.; Li, Y. Depth-Based Human Fall Detection via Shape Features and Improved Extreme Learning Machine. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1915–1922. [[CrossRef](#)]
28. Han, K.; Yang, Q.; Huang, Z. A Two-Stage Fall Recognition Algorithm Based on Human Posture Features. *Sensors* **2020**, *20*, 6966. [[CrossRef](#)] [[PubMed](#)]
29. Chen, W.; Jiang, Z.; Guo, H.; Ni, X. Fall Detection Based on Key Points of Human-Skeleton Using OpenPose. *Symmetry* **2020**, *12*, 744. [[CrossRef](#)]
30. Kostadinov, S. Understanding GRU Networks. Available online: <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be> (accessed on 26 February 2021).
31. Sharma, S. Activation Functions in Neural Networks. Available online: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> (accessed on 26 February 2021).
32. Gomez, R. Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and All Those Confusing Names. Available online: https://gombru.github.io/2018/05/23/cross_entropy_loss (accessed on 26 February 2021).
33. Ghoneim, S. Accuracy, Recall, Precision, F-Score & Specificity, Which to Optimize on? Available online: <https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124> (accessed on 26 February 2021).
34. Chollet, F. *Deep Learning with Python*, 1st ed.; Manning Publications Co.: Greenwich, CT, USA, 2017; pp. 160–166.
35. Keras Applications. Available online: <http://keras.io/api/applications> (accessed on 26 February 2021).
36. Lu, N.; Wu, Y.; Li, F.; Song, J. Deep Learning for Fall Detection: 3D-CNN Combined with LSTM on Video Kinematic data. *IEEE J. Biomed. Health Inform.* **2018**, *23*, 2168–2194.