A Robust and Automated Vision-Based Human Fall Detection System Using 3D Multi-Stream CNNs with an Image Fusion Technique

Thamer Alanazi 1, Khalid Babutain 2 and Ghulam Muhammad 1,*

1 Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; thamermatar@hotmail.com (T.A.)
2 Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
* Correspondence: ghulam@ksu.edu.sa

Abstract: Unintentional human falls, particularly in older adults, can result in severe injuries and death, and negatively impact quality of life. The World Health Organization (WHO) states that falls are a significant public health issue and the primary cause of injury-related fatalities worldwide. Injuries resulting from falls, such as broken bones, trauma, and internal injuries, can have severe consequences and can lead to a loss of mobility and independence. To address this problem, there have been suggestions to develop strategies to reduce the frequency of falls, in order to decrease healthcare costs and productivity loss. Vision-based fall detection approaches have proven their effectiveness in addressing falls on time, which can help to reduce fall injuries. This paper introduces an automated vision-based system for detecting falls and issuing instant alerts upon detection. The proposed system processes live footage from a monitoring surveillance camera by utilizing a fine-tuned human segmentation model and image fusion technique as pre-processing and classifying a set of live footage with a 3D multi-stream CNN model (4S-3DCNN). The system alerts when the sequence of Falling of the monitored human, followed by having Fallen, takes place. The effectiveness of the system was assessed using the publicly available Le2i dataset. System validation revealed an impressive result, achieving an accuracy of 99.44%, sensitivity of 99.12%, specificity of 99.12%, and precision of 99.59%. Based on the reported results, the presented system can be a valuable tool for detecting human falls, preventing fall injury complications, and reducing healthcare and productivity loss costs.

Keywords: fall detection; deep learning; 3D-CNN; multi-stream CNN; human segmentation; image fusion

1. Introduction

Falls in older adults can have far-reaching consequences, which can lead to either severe injuries or death if they do not obtain medical assistance immediately. Human falls are usually unplanned and involve falling from a higher level, when sitting or standing, to a lower level on the ground. According to [1], falls have been described by the World Health Organization (WHO) as the leading cause of trauma in elderly individuals, which indicates that approximately 30% of individuals aged over 65 years have at least one trauma event annually [2]. A total of 47% of older adults who fall lose their independence and must rely on others for their daily activities [3]. This information shows that there is a need to assist older adults as soon as they fall. In this case, it is essential to have technologies that can detect falls among this population to be able to act as soon as possible. With the rapid development of video monitoring, surveillance, and communication technologies, it is becoming increasingly feasible to detect falls immediately after they occur.
There has been increased interest in fall detection, resulting in the development of many technologies that can be used to detect these events [4,5]. Acceleration and vibration sensors [4,5] are utilized to identify human motion, sound, and vibration [5–7]. Most of the proposed methods cannot perform as expected due to several challenges, including noise that affects the acoustic sensors’ functions. Additionally, some strategies, such as floor vibration, are only possible if there are sensors on the ground. Another major challenge is that using sensors in large areas can be extremely expensive. These challenges are the reason why some people have embraced smartwatches and smartphones since they have equally effective sensors.

Other models used in the detection of falls include those that focus on gathering data from sequences of videos [8]. This strategy includes the use of multiple [9], single, omnidirectional [10], and stereo pair [11] cameras. Since cameras can record detailed information about the subject’s mobility, the method is thought to be useful in identifying whether a fall has occurred. In this situation, everyone agrees that the camera’s data are more comprehensive and insightful than those of regular sensors. However, vision-based fall detection may save lives and considerably save medical costs for the elderly.

Deep learning encompasses a wide range of methods, including hierarchical probabilistic models, artificial neural networks, and various algorithms of supervised and unsupervised feature learning [12]. In comparison to other machine learning techniques utilized in different fields in the past, deep learning approaches exhibit a superior performance [13,14]. Deep learning is unique since they have multiple layers that allow these systems to obtain data from different abstraction levels. They can comprehend large amounts of complicated data and provide useful details. Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Long-Short-Term Memory (LSTM) [15] are widely used deep learning models that outperform state-of-the-art (SOTA) techniques employed in visual processing, audio, neural language, and other sensor-based challenges. Deep learning has demonstrated a considerable leap in computer vision applications, such as object detection, activity recognition [15], semantic segmentation [16], and motion tracking.

A video is a series of still images or frames played in rapid succession to provide the impression of uninterrupted motion. CNNs find widespread use in areas such as video analysis, categorization, person identification, and posture estimation. However, incorporating the temporal dimension of videos can be challenging, which can be overcome in several ways. One approach is using 3D CNNs, which can capture the movement of objects in videos by applying a 3D filter to a sequence of frames used as input for the convolution process. This approach allows for the integration of temporal and spatial details in the same convolution operation.

One of the most important parts of object identification is identifying the kind of item present in an image or video and pinpointing its exact location with the help of a bounding box [17]. However, this method poses challenges that can be addressed through the use of deep learning techniques. Popular frameworks that are utilized include Deep Belief Networks, CNNs, and Recurrent Neural Networks (RNNs), which are commonly used in processing temporal signals.

Systems that rely on visual understanding need segmentation modules. Since the discovery of the computer vision field, image segmentation has been a primary challenge related to the separation of images into several segments [18]. Segmentation denotes a method that involves allocating labels to every image pixel, based on the understanding that images sharing a label are generally linked to the same object. Several objects may be included when segmenting an image, such as buildings, dogs, cars, or people. When image segmentation is accurate, understanding scenes becomes simpler [19].

As a computer vision problem, human segmentation has been used in numerous applications, including understanding human movement, retexturing and classifying clothing, or identifying pedestrians. Human segmentation is a crucial phase of surveillance
cameras’ functions before any recognition decision. However, this is often challenging because the human body shape, pose, environment, and clothing differ from one instance to the next. To successfully segment the human body from its background, the human segmentation approach needs the capability to define the boundaries of the human body and mask them.

The image segmentation task can broadly be handled using two methods. The first technique identifies similar image pixels from the segment. This detected similarity depends on pixel threshold values and can be attained in ML through clustering algorithms. The second technique uses the image’s pixel discontinuity. Some methods for detecting the edge, point, and line employ the discontinuity approach to obtain intermediate segments. The ultimate segmentation image can be obtained by further processing the intermediate segments. To date, the available algorithms designed to solve the image segmentation problem apply varying techniques based on neural networks, clustering, region, edge, and threshold. This work mainly solves the semantic segmentation problem using a region-based technique [20].

In semantic segmentation, pictures are categorized pixel by pixel such that each pixel belongs to a unique class cluster [21]. With the advent of deep learning, semantic segmentation has become a pivotal area of computer vision and image processing; it has experienced major research efforts and diverse applications in various fields [22].

Instance segmentation is a particular type of image segmentation that focuses on identifying objects’ instances and outlining their boundaries. It has broad practical use in various real-world situations, such as self-driving vehicles, medical imaging, and monitoring crops from above. When there are several objects of the same kind that need to be tracked individually, instance segmentation is useful.

Image fusion combines multiple images from different sources into a single image containing information from all of the input images. Image fusion aims to create a new image with improved features compared to the individual input images. This can include improved resolution, increased contrast, or reduced noise. The technique is used in many fields, including remote sensing, medical imaging, and surveillance.

Deep learning has been used for image fusion to improve the performance of the fusion process. The utilization of CNNs enables the extraction of characteristics from the input images, which can then be combined through a fusion process. This approach has been shown to improve image fusion performance, especially in cases in which there is significant variation in the input images.

Image fusion is a process that involves gathering significant details from multiple images to produce a reduced set of images, typically just one image. The resulting image is more precise and instructive than any individual image source since it contains all the essential details. The primary objective of image fusion is to have fewer images with more information and to produce images, which are easily interpretable for both humans and machines [23–25]. When various images are combined into a single image to include relevant information, it is referred to as multisensory image fusion in computer vision [26]. An image created by fusing many images contains more information than any of the individual images used to create it [27].

This paper proposed a new human fall detection system using four-stream 3D CNNs. The major contributions of the system are as follows. Each of the system’s streams is associated with one of the four stages of human decline (standing or walking, falling, fell, and at rest). Each stage consists of a series of frames with a certain orientation(s). The sequences are organized into four groups based on these characteristics. Phase-wise frame feeding to 3D convolutional neural networks is a novel idea for detecting human falls. In other studies, multi-stream CNNs have been utilized, but just to represent frames in a variety of ways, not to distinguish between stages. The poor accuracy of these systems may be traced back to the fact that some earlier efforts simply fed the CNN a set number of frames without paying attention to the semantic information (human fall phases) included within them.
The remaining parts of this paper are organized into five main sections. Section 2 reviews previous studies in the field of human fall detection, analyzing different approaches to the problem and identifying gaps in knowledge. In Section 3, the proposed methodology for the human fall detection system is detailed, highlighting the key features and their functions. Section 4 presents an in-depth analysis of the conducted experiments that evaluate the efficacy of the proposed system. The results are discussed in Section 5, which compares the proposed method’s performance with that of other approaches. Finally, Section 6 concludes the paper, summarizing the main findings, limitations, and potential avenues for future research in related fields.

2. Related Works

Human fall detection systems typically fall into two categories [28]: those that involve wearable sensors and those that rely on machine vision-based techniques. This section will cover a review of the relevant literature, exploring studies conducted on both categories within human fall detection systems.

2.1. Wearable Sensors

As our main focus is on vision-based sensors, in this section we briefly discuss some major contributions of wearable sensors in human fall detection systems. The systems use a variety of sensors, including accelerometers, pressure sensors, gyroscopes, tilt switches, and magnetometers, to record data about the user’s body movements that may be used to identify and prevent falls [29]. Additionally, these sensors can be utilized in diverse environments and are dependent on datasets to achieve accurate fall detection [30]. However, they are not easy to use for most people since they are uncomfortable when worn for long periods, making them somewhat intrusive. At the same time, it is inconvenient for most people to wear them all day long. Moreover, battery life is limited, which may be more problematic for elderly people. The other major disadvantage of wearable sensors is that they are inaccurate in classifying the falls detected, which depends on the sensor’s placement [2].

In [31], a fall detection system that uses a wrist-worn device was presented. There was a proposal for a wearable fall detection system in a smart household environment [32]. The phases in the fall detection system consisted of the start, impact, posture, and aftermath [33]. The system could generate alarms when the subject was unable to recover after a fall. There was a proposal for a low-cost, fall detection environment-aided living scheme in [34] meant for handicapped people and the elderly. The system had vibration detection sensors that could differentiate footprints, movement, and unusual action.

In [35], falls were detected using a cloud based on the Internet of Things (IoT) [36,37], along with a CNN model, CNN-3B3Conv. Notably, the system did not use any images for analysis but instead utilized sensors that played the role of accelerometers on the user’s end. These sensors were linked to a smartphone and a smartwatch to the user’s body. On the other hand, Chelli et al. [38] used data from accelerometer sensors to measure the rotation velocity for fall detection.

2.2. Vision-Based Sensors

Machine vision-based systems, unlike wearable systems, do not require users to wear anything, making them less intrusive and potentially more efficient. These systems have been applied in various fields, including human fall detection [37,39,40]. Yazar et al. [34] proposed both active and passive ambient fall detection systems and evaluated their feasibility based on factors such as power connectivity, affordability, obtrusiveness, installation environment, and complexity of installation. In [41], an ambient-assisted living system that utilized a 3D range camera to detect human falls via time-of-flight technology was presented. The camera’s field of view was not impacted by shadows or lighting, and
the system segmented elderly silhouettes from that data. An extensive dataset of falls experienced by people while leading their normal lives was analyzed to confirm the system’s efficacy.

Range-Doppler radars, also known as frequency-modulated continuous wave radars, have been used with deep learning algorithms to detect falls in elevation [42]. Time-frequency and range features were combined to improve the output of the system. The radar signal offers advantageous properties, such as providing unimpeded lighting and sensing without invading users’ privacy, which contributes to the system’s widespread adoption [43]. To identify the Doppler effect, one has to observe frequency shifts in radar readings brought on by human motion (through radar backscatter). Human falls at an assisted living facility may be detected, categorized, and located by radar. Nevertheless, radar has a number of drawbacks that need to be addressed before it can be widely used, including a high false alarm rate, a limited radar range of vision, and the obscuring of falls by big fixed objects.

In their research, the authors in [44] put forth a system for identifying falls by monitoring the spatiotemporal context of a 3D depth image taken with a Microsoft Kinect sensor. Assuming that only one elderly person was present, they utilized a Single Gaussian Model (SGM) to determine the person’s head position and the coefficients for the floor plane. The system was equipped with an algorithm that followed the target’s movements in relation to the floor. The team recorded real-life video sequences with a Kinect sensor in a simulated environment for their experiments. In another proposal [45], a low-cost vision-based system was introduced, which could identify falls causing wounds and was insensitive to changes in illumination and occlusion.

In the fall detection system proposed by [46], RGB and in-depth data were extracted using a Microsoft Kinect camera. The system was capable of successfully capturing each frame from the camera, and it also included an event for ground segmentation achieved by dynamically subtracting the background. Similarly, Stone and Skubic [47] used a Microsoft Kinect camera to detect falls in a household environment through a two-step process. The first step involved tracking the individual of interest for a period of time to obtain time-series features. The succeeding step utilized a decision tree ensemble to predict potential falls based on the sequential data collected.

In their study, the authors of [48] utilized a Microsoft Kinect sensor to identify older adults by capturing 3D depth images. To achieve target and background processing, a median filter was applied. By subtracting frames in the background from the picture, a method was used to view the individual’s silhouette as they moved through the images. The creation of a disparity map using depth images required vertical and horizontal projection histogram data. In [49], the authors discussed the process of identifying falls using a 2D CNN. The system’s uniqueness lies in its ability to recognize activities from photos captured with a Microsoft Kinect camera. The researchers demonstrated that indoor conditions could be utilized to create a new dataset. This system can detect falls by considering posture classification provided by a 2D CNN, depth information, and background elimination on RGB-D.

Several studies have employed machine learning techniques to detect human falls. For instance, in [50], a vision-based approach for detecting falls was discussed, which relied on the temporal gradualness norm. Similarly, the system in [51] employed a DeeperCut model to generate features by extracting the human skeleton and decomposing it into five parts, including two legs, two arms, and one head. The five parts generated 14 data points, which were used as input features for a deep classifier.

In [52], a fall detection system based on computer vision was proposed, which was unique from previous systems as it incorporated non-vision styles, such as the use of accelerometers. The primary goal of this study was to create a depth-camera-based vision system with improved speed, precision, and reliability, without relying on RGB pictures or requiring excessive processing complexity. An SVM was used to identify falls, and in order to limit false positives, it was trained using data from free fall body models. The
system was highly effective, as it did not produce any false positives, and the number of falls not detected was significantly low.

A multiple-stage fall detection system using computer vision was demonstrated in the research study described in [53]. The stages include extracting frames from a video, locating humans in the frames, extracting features PCANet, and classifying the fall events using an SVM classifier. On the other hand, in [54], background subtraction was first achieved with a Gaussian distribution. Using Gaussian distributions, each pixel was modeled so that variations in illumination could be handled as well as static objects placed or removed. In order to identify the foreground objects, highlighted pixels were used, and the image frame was then smoothed with post-processing to eliminate lonely pixels and fill holes in the background.

Several researchers have attempted to address the challenge of detecting falls. In one study [55], a fall detection method was proposed that relied on several scene characteristics. The author emphasized the importance of improving accuracy while minimizing incorrect classifications. Another study [56] proposed an image-based fall detection method that could be useful in nursing homes, focusing on accidents that occur when sitting on a chair or standing. The first part of this procedure entails utilizing an object detector to find the person in the picture and then determining where they are sitting in relation to the chair and the rest of the room. A program takes into account the vertical and horizontal distances between the chair and the occupant to calculate the likelihood of a fall. However, this approach only utilizes one image and does not take into account other critical factors, such as speed, body dynamics, or movement.

In [57], a dataset was created by utilizing four cameras with various angles and settings, which was then used to train a model. The proposed model involves using a median filter to eliminate the background and then applying morphological techniques to segment the area. Fall detection is achieved by combining the segmented information with the velocity of the head’s movement, the center of gravity’s speed of movement, and the person’s proportion to the image’s size. These results are fed into an SVM classifier to determine the risk of falls. The SVM model serves as the foundation for this study [58] and is used to achieve an optimal separation among the classes. However, there is a limitation in the generalization of new samples due to the insufficient feature description in the trained dataset [59]. By using a median filter, Muaz et al. [60] were able to improve the technique by eliminating the background.

Recently, researchers have been focusing on using deep learning for human fall detection. In [61], a fall detection system based on a three-streamed 3D CNN network was proposed. The pre-trained models from ImageNet were used for each stream, which was trained separately using Optical flow, Visual Rhythm, and pose estimation features. An SVM was utilized in the classification, and the URFD and Le2i FDD datasets were used. The datasets were split into 65% training, 15% validation, and 20% testing. Another study Ref. [62] used 3D CNN to develop a fall detection system. Initially, the input to the 3D CNN was the frames’ sequences to extract 3D features, which were then converted into 2D ones using a reshape layer. Multi-scale features were generated using a spatial pyramid and fed into the movement tube regression layer, tube anchors generation layer, and SoftMax classification layer. Tube anchors that matched the ground truth were found using a matching and hard negative mining layer, and falls were detected using SoftMax and the movement tube regression layer. A large-scale spatial–temporal (LSST) dataset was generated, along with the Le2i FDD [63], and data augmentation was achieved by altering the spatial, brightness, and temporal properties. From the 3D-CNN model, 16 frame sequences were used to produce 3D features, followed by tube anchor generation and SoftMax classification layers, respectively.

A recent study [64] aimed to improve the accuracy of human fall detection by developing a 3D CNN- and LSTM-based system that utilizes multiple cameras. The researchers found that using multiple cameras was more effective to capture videos than a single camera. The system also incorporates an LSTM-based mechanism and 3D CNN to identify
activities within the video footage. To address the limitations of auxiliary equipment-based systems and vision-based systems, the system uses deep learning and activity features to create a more accurate human fall detection system [65]. The system successfully performs scene analysis using the Faster Region-based Convolutional Neural Network (R-CNN) deep learning method, which can differentiate between human beings and furniture, such as a sofa. In another approach, researchers [66] proposed a deep learning method that uses either single or multiple camera setups. They used the improved visual background extractor (IViBe), which is an enhanced version of the visual background extractor (ViBe) algorithm, to implement background subtraction and obtain a silhouette of the human. These data generate a history image as well as a dynamic image, which were then used as input into a three-streams of CNN in corresponding sequences. Although this strategy was computationally heavy, it showed balanced sensitivity and specificity compared to other systems.

Another fall detection system proposed by researchers [67] uses low-resolution thermal sensors and recurrent neural networks (RNNs). Sensors were placed in pairs, with the first at 10 cm and the second at 1 m above the ground. The structure involved using a 1-dimensional deep learning convolution layer with a Rectified Linear Unit (ReLU) activation layer, a max pooling layer, RNN layers, and a dense layer. The output of the model indicates a fall if it is 0, and no fall if it is 1. Due to the challenge of using a supervised classification algorithm to assess the possibility of a person falling off an object, fall detection is framed using anomaly detection as a semi-supervised learning-based strategy [68]. One of the approaches used that can understand temporal and spatial characteristics is the deep spatiotemporal convolutional autoencoder. Researchers have conducted studies and tests on depth and thermal cameras with non-invasive sensing modalities. Although the use of autoencoders is advantageous when handling falls as an abnormality, the model’s main drawback is its boosted computational ability during the construction phase. Optimal metrics can be obtained from pictures taken by depth cameras, making this model one of the most reliable fall detection systems.

In their seminal work, Gu et al. [69] introduced a novel dataset that included various categories of activities, including falls. The researchers also developed an algorithm that combines data from two different channels based on previous studies [70], using the RCNN [71] model. The first channel is an inception of Inflated 3D ConvNet (I3D) that processes RGB images, while the second channel utilizes flow inputs that carry optimized smooth flow information, which is processed by optical flow data [72]. The researchers emphasized the need to use keyframe marks to highlight the dataset’s novelty and the importance of combining multiple channels to improve fall detection accuracy. Overall, the method proposed in [69] is considered innovative and effective in detecting falls. However, some factors can affect the system’s performance, such as the quality of the sample, which can be uneven and limit the dataset. While the dataset contains digital recreations of images obtained from movies, it can also include real photos captured by security cameras. However, the use of specific filters to process such images can lead to a loss of naturalness in the situation. In contrast, Fan et al. [73] presented a new fall detection strategy that focuses on determining the actual duration of a fall by detecting the four phases of standing, falling, fallen, and not moving. The authors used dynamic images, which provide more details, including appearance and temporal features. To achieve this, they trained a Deep ConvNet on various fall sequences using a single image based on the frames of the training video. This approach was only possible after merging the frames into a single image, and a sliding window approach was used to detect the four fall phases accurately.

Other researchers focused on exploring the use of optical flow and multi-stream fall detection methods, such as [74–77]. For example, in [74], a generic CNN model was trained on optical flow images and fed to a fully connected neural network for fall classification. Similarly, Carneiro et al. [76] used three independent CNNs, each with optical flow, RGB, and pose inputs, ensembled for improved fall detection accuracy. Leite et al.
[77] proposed a system that used optical flow, saliency map, and RGB as inputs for three independent VGG-16 networks. Each stream was classified and weighed using Dual SVM, with pre-training on ImageNet and fine-tuning on UCF101. In [78], a CNN model with fewer parameters was used to create an optical-flow-based fall detection model that could be implemented on low-computing devices, such as mobile robots. An enhanced dynamic optical flow-based fall detection system was demonstrated in [79], which used URFD, MCFD, and Le2i FDD benchmark datasets. The pre-processing phase involved converting the input video into RGB images, calculating optical flow, rank pooling, and dynamic flow to create a dynamic image. Further feature extraction and classification were performed using a modified pre-trained VGG-16 model, with the addition of a variation light to make fall detection more challenging. In [80], a 3D-CNN vector-based fall detection method was described based on a fall motion mixture model from several sequential frames, using a pre-trained ResNet-101 based multi-label dataset. Berlin and John [81] proposed a Siamese neural network with two symmetrical convolutional networks for fall detection using single-shot classification. Stacked RGB and optical flow features were used with a conventional 2D convolutional filter and depth-wise convolutional filter to generate a distance value in $[0, 1]$, indicating the similarity between the input video sequence and the ADL video sequence.

This literature review reveals that deep learning has emerged as a prominent approach in fall detection systems. However, there is a lack of clarity on how multi-stream CNNs can be utilized to distinguish between different phases of falls. There is a clear need for an effective fusion network that can efficiently combine the features extracted from various phases. Furthermore, while prevailing multi-stream CNNs have been utilized in encoding fall videos, they have not been tailored to specifically address fall phases.

3. Our Proposed Method

In recent years, deep learning has gained significant attention as a powerful tool in various applications of image and video processing, including the challenging task of human fall detection. Figure 1 shows a block diagram of a human fall detection system that uses deep learning. Various sensors, including wearable smartwatches, body sensors, and mounted IP cameras, can be utilized to gather data about the patient [29]. The obtained data are conveyed to a local server for pre-processing, which may involve filtering out redundant information, consolidating frames, and eliminating noise. Subsequently, the pre-processed data are input into a deep learning model, such as CNN. The system’s output could be standing, falling, fallen, or other activities [82]. In our proposed work, we aim to build upon our previous research [82] to develop an advanced automated vision-based method for detecting human falls using multi-streams of 3D CNNs. The complex event of human falls occurs in live or video scenes and typically involves transitions from standing (or sitting) to falling to a resting position.

Figure 1. Example of a general block diagram of a human fall detection system.
A video-level CNN model was trained to capture appearance and motion information from surveillance recordings, allowing for the collection of a wide variety of elements from people’s everyday lives. Standard 2D convolutional networks focus only on feature extraction at the frame or picture level, excluding any consideration of dynamics over many frames. In contrast, 3D CNNs are proficient in extracting features from both spatial and temporal dimensions, making them well-suited for video data processing. Furthermore, multi-stream CNNs have emerged as a new trend for handling video data, and by leveraging the capabilities of 3D CNNs, the aim is to capture both spatial and temporal features from video scenes, which is essential for robust and accurate human fall detection.

Building upon the aforementioned discussion, our proposed work created an advanced method for detecting human falls utilizing multi-stream 3D CNNs. This was accomplished by leveraging the 4S-3DCNN architecture [82], which incorporated a 4-stream 3D CNN architecture capable of analyzing videos to capture both the spatial and temporal features of human fall actions.

Unlike our previous method [82], this work employed the 4S-3DCNN model within a comprehensive fall video scenario. There were other different improvements. The current work incorporated human segmentation as a preliminary processing measure aimed at improving the accuracy of the classification process, thereby resulting in more precise fall detection. In addition to human segmentation, the method also employed image fusion on the segmented images utilizing a sequence of 32 frames instead of 16 frames. This expansion allowed for the capture of a broader scope of temporal data, further enhancing the accuracy of the method.

The proposed method, as shown in Figure 2, begins by capturing 32 consecutive frames from an input video, which corresponds to roughly one second of live video at a frame rate of 30 frames per second (fps), representing a fall action. These frames are then converted into grayscale images. Next, a fine-tuned deep learning model designed for human semantic segmentation [83] is utilized to segment human appearance within the frames. An image fusion technique is then applied to the segmented images to generate four RGB pre-processed images. Details of the pre-processing procedure, which involves human segmentation and image fusion, are provided in Sections 3.1 and 3.2, respectively.

These pre-processed images are then fed into the previously developed 4S-3DCNN model that uses a 4-branch architecture, wherein each branch concentrates on a single image’s worth of feature-learning data, gathered in a distinct order via space and time. This model classifies the 32 frames into one of the four classes that are trained on, including either Standing, Falling, Fallen, or Others. In case there is a sequence reporting Falling followed by Fallen, the method alerts the occurrence of a human fall situation. Section 3.3 provides details of classification and human fall alerts.

3.1. Human Segmentation

Human segmentation is a computer vision task that involves separating human figures or shapes from the background in images or videos. The goal is to accurately distinguish between foreground (human shapes) and background pixels in an image or video. This task has various applications, including surveillance systems, medical imaging, and video editing. Deep-learning-based approaches have achieved significant progress in human segmentation, with CNNs being the most used technique. The segmentation process is typically performed by classifying pixels as either belonging to the foreground or background using a trained neural network [83].

Gruosso et al. [83] proposed an approach for human segmentation in surveillance videos using deep learning techniques. The authors leveraged a combination of CNNs and Fully Convolutional Networks (FCNs) to extract features from the input frames and perform pixel-level segmentation. The proposed method uses an encoder–decoder neural network, developed from the SegNet [83], to classify pixels and distinguish between foreground (human shapes) and background. Figure 3 shows the architecture of the human
segmentation model. The authors also demonstrated the practicality of their approach by applying it to real-world surveillance footage and achieving accurate segmentation results.

Figure 2. The architecture of the proposed fall detection system.

Our approach involves utilizing the human segmentation model proposed by [83] as the initial pre-processing step. This model was designed to effectively segment humans from irrelevant background objects, thus facilitating the accurate classification of human actions. By leveraging this model as a pre-processing step, our method aimed to improve the overall accuracy of human fall identification. In our method, the human segmentation starts by reading an original input image, as illustrated in Figure 4A. Then, it converts the input image into a grayscale image resized to $256 \times 256 \times 1$, as shown in Figure 4B. The resulting image is then processed by a trained human segmentation CNN model, which produces an image segmented by class, as shown in Figure 4C. Morphological image analysis is then implemented to obtain the final segmentation of the human within the original image [84], as shown in Figure 4D.
Figure 4. Human segmentation. (A) Original input image, (B) converting to grayscale, (C) applying the human segmentation CNN model, and (D) keeping the most sizable cluster of pixels and filling any holes.

To implement this approach, we fine-tuned and retrained the original model of [83] on a manually selected and annotated set of images from the Le2i dataset. Each image was manually annotated with pixel-level labels, where each pixel was categorized as either a person pixel or a background. During fine-tuning, we updated the original model’s input layer to accept grayscale images of size $256 \times 256 \times 1$ and transformed the first layer’s 3-dimensional filters $f(x, y, 3)$ into 2-dimensional filters $f'(x, y)$ using the mapping $F: f(x, y, 3) \rightarrow f'(x, y)$, so that:

$$f'(x, y) = F(f(x, y, 3)) = \frac{1}{3} \sum_{i=1}^{3} f(x, y, i).$$

To ensure the reliability and generalizability of the fine-tuned pre-trained model, we employed the 3-fold cross-validation technique. Despite the high segmentation results that can be achieved through fine-tuning, it is common for specific pixels or groups of pixels to be classified incorrectly. This can result in some pixels being wrongly labeled as a person while they belong to the background, and vice versa. To fix this, post-processing is required, which entails patching up any missing pixels and removing the small-connected components of pixels while keeping the large-connected group of pixels [84,85].

### 3.2. Pre-Processing with Image Fusion

Image fusion combines multiple images from different sources into a single image containing information from all the input images. Image fusion aims to create a new image with improved features compared to the individual input images. This can include improved resolution, increased contrast, or reduced noise [26,27]. When two frames from a video sequence are fused, the aspects of motion that contribute to scene interpretation and categorization are brought to the forefront [86].

The proposed method involves taking a sequence of 32 frames from a video, converting them into grayscale, segmenting them to isolate human appearance, and resizing them to $128 \times 128$. The aim is to create four fused images that are then fed into a previously developed 4-branch 3D CNN model (4S-3DCNN), with each image being processed by a separate branch. To obtain these four fused images, a fusion process using three levels of image fusion was proposed. Initially, every odd frame is fused with the following frame, resulting in 16 fused frames. The same fusion approach was then performed for the second and third levels, resulting in eight and four pre-processed images by fusion, respectively. Figure 5 provides an example of how this three-level image fusion process generates four fused images from the 32 frames that were grayscale-converted and human-appearance-segmented.
Figure 5. Three-level frame fusion to yield four fused images out of the 32 grayscale-converted and human-appearance-segmented frames.

The formulation for the three-level image fusion is as follows:
First-level fusion for 32 images:
\[ A_1 = \text{fuse}(F_1 + F_2); \quad A_2 = \text{fuse}(F_3 + F_4); \ldots \quad A_{16} = \text{fuse}(F_{31} + F_{32}) \]
Second-level fusion for 16 images:
\[ B_1 = \text{fuse}(F_1 + F_2); \quad B_2 = \text{fuse}(F_3 + F_4); \ldots \quad B_8 = \text{fuse}(F_{15} + F_{16}) \]
Third-level fusion for 8 images:
\[ C_1 = \text{fuse}(B_1 + B_2); \quad C_2 = \text{fuse}(B_3 + B_4); \quad C_3 = \text{fuse}(B_5 + B_6); \quad C_4 = \text{fuse}(B_7 + B_8); \]

3.3. Action Classification and Fall Alert

In the final step of the proposed method, the goal is to classify the action and detect if a human has fallen, as in Figure 2. To accomplish this, the pre-processed images that have undergone human segmentation and image fusion are fed into the previously developed 4S-3DCNN model that has a 4-branch architecture, as depicted in Figure 6. Each branch within this model handles one image of the four resulting fused images to learn features that capture different spatial and temporal information. The model then classifies every 32 frames of the input video into one of four classes: Standing, Falling, Fallen, or Others, based on the learned features. If there is a sequence of Falling followed by Fallen, the method detects a human fall situation and alerts accordingly.
4. Experiments

The equipment utilized in conducting the experiments consisted of an Intel® Core™ i9-9900K central processing unit operating at a speed of 3.60 GHz and equipped with 64 GB of RAM, and it operated on a 64-bit Windows 10 operating system. Additionally, it had an NVIDIA GeForce RTX 2080Ti graphics processing unit with 11 GB of memory. The machine’s capacity to perform the experiments was verified by running a 64-bit edition of MATLAB R2022b.

The experiments conducted in this study involved assessing the effectiveness of the fine-tuned human segmentation model as well as evaluating the fall action classification of the full proposed method using the Le2i fall dataset. During the evaluations, we employed commonly used metrics to measure the performance, including accuracy, sensitivity, specificity, and precision. However, the performance of the human segmentation CNN model was assessed through various metrics, such as pixel global accuracy, mean recall, mean Intersection Over Union (IoU), and Weighted Intersection Over Union (wIoU). All these metrics were calculated using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) concepts [85], as illustrated in Table 1.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>( \frac{TP + TN}{TP + TN + FP + FN} \times 100% )</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>( \frac{TP + TN + FP + FN}{FP + FN} \times 100% )</td>
</tr>
<tr>
<td>Specificity</td>
<td>( \frac{TP}{TP + FN} \times 100% )</td>
</tr>
<tr>
<td>Precision</td>
<td>( \frac{TP + FN}{TN} \times 100% )</td>
</tr>
</tbody>
</table>

4.1. Database Description

The proposed study considered the Le2i fall detection dataset [63] that comprises fall and normal videos. The normal category contained videos with normal daily activities, such as walking, standing, squatting, and sitting. The data were obtained from fixed cameras set up at home, coffee shop, lecture hall, and office. The actors had different clothing at different sections and tried to simulate normal activities and falls to make the dataset as diverse as possible. Other features obtained included shadows, occlusions, and varying levels of illumination that were also annotated manually, especially the start and end of the frames.
4.2. Dataset Preparation and Augmentation

To fine-tune the human segmentation model, a dataset of 355 images was created by randomly selecting images from videos in the Le2i dataset. The images were manually labeled using MATLAB Image Labeler, where each pixel in the selected images was labeled as either Person or Background using pixel-wise labeling. To prevent overfitting during the training or fine-tuning of deep learning models, it is necessary to have a large number of training images. Therefore, three augmentation techniques were applied to this dataset [63]. These techniques included image translation in both x-axis (horizontal) and y-axis (vertical) directions. To utilize the translation technique, it is necessary to define a range of pixel translations. In this case, the pixel range was set at 10 pixels for both the horizontal and vertical translation directions. The third technique of augmentation was random rotation, which rotates every image in the range of 10° clockwise and anticlockwise.

To evaluate the effectiveness of fall action reporting on the proposed method, the entire Le2i dataset was analyzed and labeled. The dataset comprised a total of 177 videos, with 120 videos containing fall action and 57 videos without any fall. This comprehensive dataset was used to assess the proposed method’s ability to accurately report fall actions, ensuring that the method’s performance was evaluated under diverse conditions. Table 2 shows the preparation of the needed datasets.

Table 2. Dataset preparation for the human segmentation model and fall action video classification.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Data Type</th>
<th>Labeling Type</th>
<th>Labeling As</th>
<th>No. of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Segmentation</td>
<td>Image</td>
<td>Pixel as class</td>
<td>Person</td>
<td>355</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Background</td>
<td></td>
</tr>
<tr>
<td>Fall Action</td>
<td>Video</td>
<td>Video as class</td>
<td>Fall</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Not a Fall</td>
<td>57</td>
</tr>
</tbody>
</table>

4.3. Experimental Results

The fine-tuning of the human segmentation model was undertaken using a rigorous method that incorporated a three-fold cross-validation process to ensure the robustness and generalizability of results; in addition, data shuffling was utilized during training. Through this procedure, the model achieved remarkable metrics, including a pixel global accuracy of 98.46%, a mean sensitivity of 92.78%, a mean Intersection Over Union (IoU) of 83.98%, and a mean weighted IoU of 97.24%, as shown in Table 3.

Table 3. Results of the fine-tuning of the human segmentation model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Global Accuracy</th>
<th>Mean Sensitivity</th>
<th>Mean IoU</th>
<th>Weighted IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Segmentation</td>
<td>98.46%</td>
<td>92.78%</td>
<td>83.98%</td>
<td>97.24%</td>
</tr>
</tbody>
</table>

The implementation of three-fold cross-validation served to reduce overfitting by dividing the dataset into three distinct subsets. Each subset was used in turn as a validation set, while the remaining two subsets formed the training set. This ensured that the model’s performance was validated on different portions of the dataset, thus contributing to the generalizability of the model.

The resulting model was highly effective in distinguishing between a person and their surroundings. This effectiveness is further demonstrated by experimental examples of human segmentation using the fine-tuned [83] model on each type of falling action as well as other actions (such as sitting), with post-processing, as illustrated in Figure 7. These examples highlight the model’s accuracy in segmenting human subjects, further supporting its effectiveness and reliability, as proven by the three-fold cross-validation process.
On average, the segmentation network took 0.21 s to segment one image.

![Original | Segmented | Post-Processed | Final Result](image)

**Figure 7.** Examples of human segmentation using fine-tuning [83] with post-processing.

In the experiment with the entire proposed method for fall action classification, the prepared dataset included 120 videos, each containing fall action, and 57 videos without any falls. The method was applied to each video, and a binary result was reported based on the presence or absence of a fall situation. The classification results of the proposed method, as in Table 4, were highly accurate, achieving an accuracy of 99.44%, a sensitivity of 99.12%, a specificity of 99.12%, and a precision of 99.59%. The confusion matrix of the proposed method’s performance in classifying fall actions can be seen in Figure 8, providing a visual representation of the classification results. These impressive results indicate the effectiveness of the proposed method in accurately detecting fall actions in videos.

**Table 4.** Results of the proposed method for fall action classification.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Method</td>
<td>99.44%</td>
<td>99.12%</td>
<td>99.12%</td>
<td>99.59%</td>
</tr>
</tbody>
</table>

![Confusion Matrix](image)

**Figure 8.** Confusion matrix of the proposed method’s performance.

5. Discussion

In this section, we compare our proposed technique for human fall detection to the state of the art in terms of numerous metrics, such as accuracy, sensitivity, specificity, and precision. The comparison in Table 5 provides a comprehensive overview of the performance of various fall detection models. The accuracy of the models ranges from 78.50%
The proposed method has the highest accuracy of 99.44%, which indicates that it has a high percentage of correctly classified fall and non-fall videos compared to other models.

Table 5. A comparison of our proposed method with other studies on the Le2i dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chamle et al. [87]</td>
<td>2016</td>
<td>79.30%</td>
<td>84.30%</td>
<td>73.07%</td>
<td>79.40%</td>
</tr>
<tr>
<td>Núñez et al. [74]</td>
<td>2017</td>
<td>97.00%</td>
<td>99.00%</td>
<td>97.00%</td>
<td>-</td>
</tr>
<tr>
<td>Alaoui et al. [88]</td>
<td>2017</td>
<td>90.90%</td>
<td>94.56%</td>
<td>81.08%</td>
<td>90.84%</td>
</tr>
<tr>
<td>Poonsri et al. [89]</td>
<td>2018</td>
<td>86.21%</td>
<td>93.18%</td>
<td>64.29%</td>
<td>91.11%</td>
</tr>
<tr>
<td>Carneiro et al. [76]</td>
<td>2019</td>
<td>98.43%</td>
<td>99.90%</td>
<td>98.32%</td>
<td>-</td>
</tr>
<tr>
<td>Leite et al. [61]</td>
<td>2020</td>
<td>99.00%</td>
<td>99.00%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zou et al. [62]</td>
<td>2021</td>
<td>97.23%</td>
<td>100.00%</td>
<td>97.04%</td>
<td>-</td>
</tr>
<tr>
<td>Vishnu et al. [80]</td>
<td>2021</td>
<td>78.50%</td>
<td>98.90%</td>
<td>-</td>
<td>99.10%</td>
</tr>
<tr>
<td>Youssfi et al. [90]</td>
<td>2021</td>
<td>93.67%</td>
<td>100.00%</td>
<td>87.00%</td>
<td>83.62%</td>
</tr>
<tr>
<td>Our Previous Work [82]</td>
<td>2022</td>
<td>99.03%</td>
<td>99.00%</td>
<td>99.68%</td>
<td>99.00%</td>
</tr>
<tr>
<td>Our Proposed Method</td>
<td>2023</td>
<td>99.44%</td>
<td>99.12%</td>
<td>99.12%</td>
<td>99.59%</td>
</tr>
</tbody>
</table>

Sensitivity is an important metric for fall detection, as it measures the ability of the model to correctly identify a fall event. The models have sensitivities ranging from 84.30% (obtained by Chamle et al. [87]) to 100.00%. The models by Zou et al. [62] and Youssfi et al. [90] have perfect sensitivities of 100.00%, indicating that they can classify all fall events correctly. Our proposed method achieved a sensitivity of 99.12%, which is above average in comparison to the other sensitivities of similar works.

Specificity measures the ability of the model to identify non-fall events correctly. The range of specificities of the compared works is between 64.29% (obtained by Poonsri et al. [89]) and 98.32%. Chamle et al. [87] and Alaoui et al. [88] also have specificity less than 85%. The model proposed by Carneiro et al. [76] demonstrates the highest specificity among the compared works, with a value of 98.32%. The specificity of our proposed method outperformed others, as it achieved 99.12%.

Precision measures the number of correctly predicted fall events. The models have precision values ranging from 79.40% (obtained by Chamle et al. [87]) to 99.59%. The proposed method has the highest precision of 99.59%, indicating that it can accurately predict fall events.

Overall, our proposed model has the highest accuracy, specificity, precision, and above-average sensitivity compared to other similar methods. These results demonstrate the effectiveness of our approach to human fall detection and its potential for real-world applications.

6. Conclusions

In conclusion, this paper presented an innovative automated vision-based system for detecting falls and issuing instant alerts upon detection. By utilizing a fine-tuned human segmentation model and image fusion technique as pre-processing and classifying sets of live footage with a 3D multi-stream CNN model (4S-3DNN), the system can effectively detect when a monitored human experiences a sequence of falling followed by fallen. By applying human segmentation pre-processing, the system can isolate the human appearance in 32 input frames and use three-level image fusion to create four fused images that capture movement features across the consecutive frames. The 4S-3DNN architecture has four branches, each corresponding to one of the pre-processed images. Each branch is responsible for extracting and learning features from a different set of consecutive spatial and temporal information. The model classifies the 32 frames into one of four classes, including Standing, Falling, Fallen, or Others. If there is a sequence reporting Falling followed by Fallen, the system alerts of a human fall situation. The system’s effectiveness
was assessed using the publicly available Le2i dataset, achieving impressive accuracy, sensitivity, specificity, and precision values of 99.44%, 99.12%, 99.12%, and 99.59%, respectively. Therefore, this proposed system has the potential to be a valuable tool for detecting human falls, preventing fall injury complications, and reducing healthcare and productivity loss costs.

Despite its effectiveness in detecting human falls, the proposed method has some limitations. Firstly, it can only detect falls in scenes that have one person without any localization of the person within the video. Additionally, the experiments conducted to evaluate the method’s performance were only conducted on the Le2i fall detection dataset, which may not represent real-world scenarios accurately. In future research, we aim to explore the performance of this method on various publicly available datasets, such as MCFD and URFD, to enhance its generalizability and effectiveness in real-world settings. Additionally, we will consider integrating comparisons with wearable sensor-based methods to provide a broader perspective on the problem.

Author Contributions: Conceptualization, methodology, software, investigation, resources, formal analysis, and validation, T.A., K.B. and G.M.; data curation, T.A. and K.B.; writing—original draft preparation, T.A.; writing—review and editing, supervision, project administration, and funding acquisition, G.M. All authors have read and agreed to the published version of the manuscript.

Funding: The work was funded by Researchers Supporting Project number (RSP2023R34), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Le2i dataset [63] is used in the experiments. The dataset can be obtained from http://le2i.cnrs.fr (accessed on 26 March 2023).

Acknowledgments: The authors extend their appreciation to Researchers Supporting Project number (RSP2023R34), King Saud University, Riyadh, Saudi Arabia.

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

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.