

Article

An Image-Based Fall Detection System for the Elderly

Kun-Lin Lu ¹ and Edward T.-H. Chu ^{2,*}

¹ Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan; m10517015@yuntech.org.tw

² Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan

* Correspondence: edwardchu@yuntech.edu.tw; Tel.: +886-05-5342601

Received: 7 September 2018; Accepted: 16 October 2018; Published: 20 October 2018

Featured Application: By using image recognition and object detection, we presented the system named IFADS to detect the fall, especially the falls that occur while sitting down and standing up from a chair for nursing homes, where public areas are usually equipped with surveillance cameras.

Abstract: Due to advances in medical technology, the elderly population has continued to grow. Elderly healthcare issues have been widely discussed—especially fall accidents—because a fall can lead to a fracture and have serious consequences. Therefore, the effective detection of fall accidents is important for both elderly people and their caregivers. In this work, we designed an Image-based Fall Detection System (IFADS) for nursing homes, where public areas are usually equipped with surveillance cameras. Unlike existing fall detection algorithms, we mainly focused on falls that occur while sitting down and standing up from a chair, because the two activities together account for a higher proportion of falls than forward walking. IFADS first applies an object detection algorithm to identify people in a video frame. Then, a posture recognition method is used to keep tracking the status of the people by checking the relative positions of the chair and the people. An alarm is triggered when a fall is detected. In order to evaluate the effectiveness of IFADS, we not only simulated different fall scenarios, but also adopted YouTube and Giphy videos that captured real falls. Our experimental results showed that IFADS achieved an average accuracy of 95.96%. Therefore, IFADS can be used by nursing homes to improve the quality of residential care facilities.

Keywords: intelligent surveillance systems; elderly fall detection; object detection; image recognition

1. Introduction

Because of declining birth rates and advances in medical technology, many countries are already, or soon will be, aged societies. Thus, providing high-quality elderly care is becoming increasingly important. According to a report of the World Health Organization, falls are one of the leading causes of accidental or unintentional deaths worldwide. Each year, an estimated 646,000 individuals die from falls globally, of which over 80% are in low- and middle-income countries [1]. Therefore, the effective detection of falls is important for both elderly people and their caregivers.

Many methods have been employed to detect falls in elderly people. We categorize them into three types: wearable-device-based, environmental-sensor-based, and image-based methods. For wearable-device-based methods, most of them utilize triaxial accelerometers to detect the sudden changes of acceleration that are caused by a fall. For example, Lai et al. [2] used sensors distributed over the body to determine the injury level when a fall occurs. Similarly, Ando et al. [3] used an

accelerometer and a gyroscope in a smartphone to detect a fall. However, wearable devices are inconvenient for older adults, because they often suffer from forgetfulness. On the other hand, environmental-sensor-based methods use various kinds of sensors to sense and interpret fall accidents in the surrounding environment. For example, Feng et al. [4] used a smart floor that was embedded with pressure-sensitive fiber sensors to detect a fall. However, pervasively deploying sensors is costly and impractical.

For the image-based methods, Auvinet et al. [5] first constructed a three-dimensional (3D) shape of an elderly person using multiple cameras and then analyzed the changes in the shape along the vertical axis. Similarly, Diraco et al. [6] evaluated the distance between the centroid of a human 3D model and the ground in order to detect a fall accident. However, it is impractical and challenging to track a person continuously by multiple cameras, especially in public areas. Recently, Brulin et al. [7] first used a machine learning method [8] to identify people in a given image and then adopted a fuzzy-logic-based recognition method to detect fall events. Unfortunately, their method may not be workable when there are overlaps between an elderly person and surrounding objects in the environment.

In this paper, we propose an Image-based Fall Detection System (IFADS) to detect falls in the elderly in a timely and accurate manner. IFADS can detect falls that occur while walking forward, sitting down, and standing up, which well-represents the most common types of falls in a long-term care environment. Unlike the existing fall detection algorithms, we particularly focus on falls that occur while sitting down and standing up from a chair, because the two activities together account for a higher proportion of falls than forward walking [9]. IFADS first applies an object detection algorithm to identify people in a video frame. Then, a posture recognition method is used to keep tracking the status of the people by checking the relative positions of the chair and the people. An alarm is triggered when a fall is detected. In order to investigate the effectiveness of IFADS, we not only simulated different fall scenarios, but also adopted YouTube and Giphy videos that captured real falls. We then evaluated the effects of camera angles and multiple objects on the method's accuracy. Our experimental results showed that IFADS achieved an average accuracy of 95.96% in detecting fall accidents. We also compared IFADS with other existing image-based methods, and the results showed that IFADS is a practical solution to help caregivers, security staff, and responders quickly detect falls. IFADS can be used by nursing homes to improve the quality of residential care facilities. Furthermore, IFADS can be easily extended and applied to any place as long as there is a camera, such as smart homes, parks, and libraries.

The rest of the paper is organized as follows. Section II introduces relevant fall detection systems and their limitations. Section III describes the needs of, and challenges for, fall detection systems. Section IV describes the design and methodology of IFADS. Section V presents IFADS's accuracy through a series of experiments and real cases, and Section VI concludes the paper.

2. Related Works

Several methods have been used to detect falls in the elderly. We classify them into three categories: wearable-device-based, environmental-sensor-based and image-based methods. Each of them is described in the following paragraphs.

2.1. Wearable-Device-Based Methods

Rucco et al. [10] provided a review of the approaches that have been proposed for fall assessment, fall prevention, and fall detection that use wearable devices. According to their survey, most of the wearable-device-based methods adopted triaxial accelerometers to collect fall signals. Besides this, the review shows that the most-used segment for the wearable device sensor is the trunk of the body, since it is tightly coupled to walking. For example, Lai et al. [2] used triaxial accelerometers distributed over a body to collect acceleration signals. When the acceleration exceeded a pre-defined threshold, the system determined that it was the result of a fall accident. Their method could also determine the injury level by comparing the acceleration at impact with normal acceleration. Similarly, Tong et al. [11], Aberyuan et al. [12], and Pannurat et al. [13] used

machine learning technologies to analyze signals that were received from triaxial accelerometers that were distributed over a body. In addition, Liu et al. [14] detected falls by using not only acceleration information but also angular velocity information. The accuracy of wearable-device-based methods can be improved by obtaining signals from different kinds of sensors. For example, Lustrek et al. [15] used location sensors to locate the user. When the user lay on the ground, the system triggered an alarm. Pierlenoi [16] and Sabatini et al. [17] used not only triaxial accelerometers but also gyroscope, magnetometer, and barometer sensors to recognize the posture of users. Ejupi et al. [18] developed a method to identify users at risk of falls and maintain daily tracking. Similarly, Ando et al. [3] used the accelerometers and gyroscopes in smartphones to detect falls and track users' pathology during rehabilitation tasks. However, the elderly usually forget to wear wearable devices and smartphones. Thus, the existing wearable-device-based methods are not practical for long-term use.

2.2. Environmental-Sensor-Based Methods

Several research efforts have been made to detect falls by environmental sensors, such as pressure, acoustic, and radar sensors. Feng et al. [4] used a smart floor that was embedded with pressure-sensitive fiber sensors to detect fall events by feature-specific pressure images containing motion features for human activity analysis. The system could be set up indoors, even in the bathroom. Similarly, Dather et al. [19] used force sensors and accelerometers that were concealed under intelligent tiles to locate, track, and recognize user activities, such as walking, standing, sitting, lying down, and falling. However, the proposed method was costly and could only be used indoors.

Li et al. [20] proposed a method, named acoustic-FADE, that detected falls by acoustic sensors. The system consists of a circular microphone array that can capture sounds indoors. When a sound is detected, the system locates the source, enhances the signal, and classifies the event as a "fall" or a "non-fall" by machine learning approaches. However, it is unrealistic to set up a circular microphone array in each and every room. In addition, the classification becomes challenging when there is more than one person in the room.

Su et al. [21] detected human falls by using a ceiling-mounted Doppler range control radar. Due to the Doppler effects, the radar senses any motions from falls as well as non-falls. They used wavelet transform (WT) coefficients at several scales over continuous frames to form a feature vector for the "fall" versus "non-fall" classification. Similarly, Shiba et al. [22] proposed a system based on a microwave Doppler sensor. The system calculates the trajectories of the frequency distributions that correspond to the velocities of the movements while falling and classifies the events by a hidden Markov model. However, a signal's analysis can be easily affected by the number of people inside the room.

Wang et al. [23] designed WiFall, which takes advantage of the physical layer channel state information (CSI) in WiFi infrastructure. WiFall can also effectively recognize common daily activities, such as walking, standing up, sitting down, and falling. Wang et al. [24] proposed a similar method, named RT-Fall, to exploit CSI in WiFi devices to improve WiFall. However, neither WiFall nor RT-Fall can work if there is more than one person in the room. Kido et al. [25] used thermal imaging sensors to differentiate between the normal activity and the falling activity in a toilet room where the temperature should be less than 31 °C. However, the applicability of the proposed method may be limited due to privacy issues.

In summary, the accuracy of environmental-sensor-based methods can be easily affected by objects in the environment. In addition, pervasively deploying sensors is costly and impractical.

2.3. Image-Based Methods

Most of the image-based methods can be categorized into four types: multi-camera, single-camera, depth-camera, and wearable-camera methods. For multi-camera methods, Auvinet et al. [5] reconstructed 3D shapes of people using multiple cameras and detected falls by analyzing the

volume distribution along the vertical axis. However, it is hard and impractical to track people using multiple cameras in public areas.

For single-camera methods, Brulin et al. [7] proposed a posture recognition method based on fuzzy logic. In order to detect a person among other moving objects, they adopted a machine learning technology to recognize a person [8]. Yu et al. [26] likewise proposed a posture recognition method by analyzing the human silhouette. In order to identify a human body among moving objects, the object with the greatest number of moving pixels is regarded as a human body. Then, ellipse fitting and a projection histogram are used as the global and local features, respectively, to describe different postures. Finally, a directed acyclic graph support vector machine (DAGSVM) is used to classify the posture. However, if the person lies on the ground, the method may issue a false alarm. Mirmahboub et al. [27] used two different background separation methods to find a human silhouette and used the area of the silhouette as a feature to feed into a support vector machine (SVM) for classification. Agrawal et al. [28] used background subtraction to find objects in the foreground and categorized a human by contour-based human template matching. They detected a fall by computing the distance between the top and mid center of the bounding box of a human. Poonsri et al. [29] adopted a background subtraction and Gaussian mixture model to detect human objects. They then computed the orientation, aspect ratio, and area ratio to extract features and classify the postures. However, background subtraction may not be able to correctly detect multiple human objects and results in an inaccurate classification result. Furthermore, if the person is obscured by other objects, the above mentioned methods may not work properly.

For depth-camera methods, Ma et al. [30] proposed a method that involves extracting curvature scale space (CSS) features of human silhouettes from each frame and representing the action by a bag of CSS (BoCSS) words. They then identify the BoCSS representation of a fall from those of other actions by utilizing the extreme learning machine (ELM) classifier. Bian et al. [31] proposed another method to detect the motion of a fall by using the SVM classifier with the input of the 3D trajectory of the head joint. Diraco et al. [6] evaluated the distance of a 3D human centroid from the floor plane to detect falls. Angal et al. [32] used the Microsoft Kinect sensor to detect a fall by collecting information on the velocity, acceleration, and width height ratio of a human object. However, the above-mentioned methods may not be accurate when the person lies on the ground. In addition, these methods cannot recognize the difference between falls and sitting on the ground. In summary, all the above-mentioned methods focus on falls that occur during forward walking. They cannot accurately detect falls that occur while sitting down and standing up from a chair.

For wearable-camera methods, Ozcan et al. [33] used a reverse approach by taking a completely different view compared with the existing vision-based systems. The system employed a modified version of the histograms of oriented gradients (HOG) approach together with gradient local binary patterns (GLBP), and it detected falls by the change in the dissimilarity distances obtained by HOG and GLBP. However, the wearable-camera methods are not practical for daily usage, because the elderly usually suffer from forgetfulness.

3. Design Requirements and Challenges

An IFADS for the elderly should fulfill four design requirements: high accuracy, automation, low-cost, and real-time computing. To ensure high accuracy, there should be an accurate and comprehensive way to detect falls in the elderly and then help the elderly recover well. In addition, manually monitoring surveillance images is extremely inefficient and slow; so, an automated technique is needed. Third, the hardware cost should be low. Finally, to help the elderly receive medical care in time after a fall, the computation time for fall detection should be short. The challenges of designing such a system are described below.

For image-based methods, the major technical difficulty lies in the fact that it is difficult to detect a falling person. When a person falls, the person may be obscured by other people or surrounding objects. As a result, the camera may not see the person or only see parts of the person. On the other hand, it is very difficult to identify human features when a person falls. Another challenge is recognizing the posture of the person. Most of the existing surveillance systems do not

provide in-depth information, so it is difficult to recognize the posture from two-dimensional (2D) human shape information, especially in public areas. Therefore, the accurate detection of a falling person is a challenge. For this, we propose a method to track the state of the person continuously. For the situation where the person is invisible or partially visible, our method can determine a fall by backward tracking from the last few frames.

Environmental obstacles and similarities in colors are also challenges in correctly determining that a fall event has occurred. In order to eliminate the effect of environmental obstacles on accuracy, we use status tracking to evaluate the relationship between the person and environment obstacles. If the person suddenly disappears from the scope of the surveillance system and his/her previous status is not leaving the edge of the surveillance area, we consider this situation as a fall event in which the person is obscured by obstacles. On the other hand, if the person is partially obscured by obstacles, but he keeps walking and his/her height does not change, we consider this situation as normal (Please see Section 4.4). In order to address the problem of similarities in colors, we adopted YOLO, a well-known object detection method, to detect human objects sufficiently. Since YOLO uses not only color information but also contour information to discover human objects, the effect of color similarity can be significantly reduced (Please see Section 4.2).

4. Methodology

In this work, we present an IFADS for the elderly to detect a falling person. IFADS can be easily integrated into existing surveillance systems and web cameras to provide caregivers with real-time information. IFADS has five functions: object detection, person tracking, person positioning, posture recognition, and fall detection. In the following subsections, we describe the system architecture and operating flow (Section 4.1), object detection and person tracking (Section 4.2), person positioning and posture recognition (Section 4.3), and fall detection (Section 4.4).

4.1. System Architecture and Operating Flow

As it is a complete surveillance system, IFADS is able to detect falls in the elderly by using images from surveillance cameras in public areas to offer them prompt treatment. In addition to falls during walking, we focus on falls that occur while sitting down and standing up from a chair, because the two activities together account for a higher proportion of falls than forward walking.

Figure 1 shows the architecture of IFADS. First, when the person appears on camera, the person is far from the chair. IFADS recognizes that the person is standing. If the change in the height of the person's bounding box is significant, it will further determine whether the person has fallen. In addition, if the person is beside the chair, since he/she may sit on the chair, IFADS uses the posture of the person to determine their status. When the person is trying to sit down, their status is in progress. In addition, if the person sits on the chair, their status is sitting. Since the person may fall while sitting down or standing up from the chair, IFADS checks if the person is in danger. Finally, if IFADS cannot detect the person, it will use the person's previous state to make a decision.

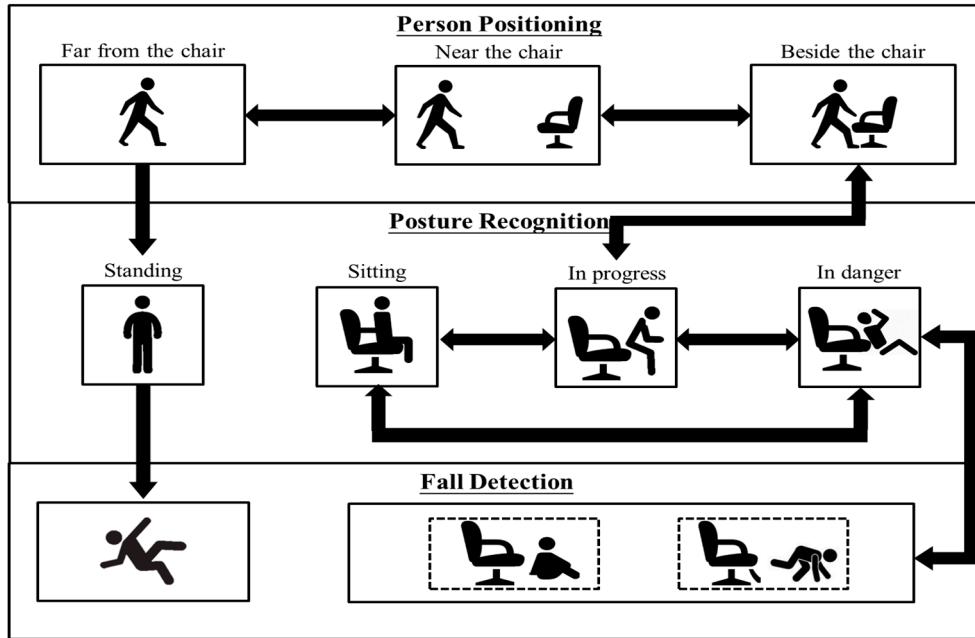


Figure 1. System architecture of the Image-based Fall Detection System (IFADS).

We adopted the well-known human model [34,35] to determine a body posture. As shown in [35], the upper portion of the body from the crotch to the head is four head lengths. Also, the lower portion of the body from the feet to the crotch is four head lengths. The thigh is two head lengths, the knee is one head length, and the calf is two head lengths. Based on the human model, we take the height changes while walking to determine the body posture. If the height decreases more than a half of a head length, the posture is regarded as the starting point of sitting down. In addition, if the person is falling down or kneeling down, the length below the navel (five head lengths) should reduce by over half. Therefore, the height will be less than $5.5/8$ ($= (3 + 2.5)/8$) of the original height. Finally, for the case where the person suddenly disappears from the scope of the surveillance system and his/her height in the previous frame is less than the average height of standing up and kneeling down (i.e., $6.5/8$ ($= (8 + 5.5)/2$)), he or she is regarded as falling down.

4.2. Object Detection and Person Tracking

The purpose of object detection and tracking is to identify people in a given video and track their movement. For this, IFADS first utilizes YOLOv3 [36] to detect the person, the chair, and the bench, and then uses the object tracking method continuously adaptive mean shift (Camshift) to track the person continuously. The notations and definitions used in the paper are shown in Table 1. Algorithm 1 shows the steps for object detection. First, IFADS extracts a frame F from a video stream. Second, since the YOLO network downsamples the input by 32, we fill the image size of F to a multiple of 32. Third, we load the model parameter of YOLOv3 to detect the object. Fourth, the bounding box O_i is the result that is detected by the YOLO detector. Finally, for a box O_i , if it detects O_i as the chair or the bench, it names it C_i . If it detects O_i as the bench, for a bench with no seatback, we consider that the height of the bench is twice that of a bench with a seatback. Conversely, if IFADS detects O_i as the person, it names it P_i . However, people may be obscured by the chair or the bench when they are walking around. Since the location of the chair or bench in a public area, such as a park or a courtyard, is stable, we set the position and the size of the chair or bench. Since there are usually many people in public areas, we use the object tracking method called Camshift to track the person in P_i .

Table 1. List of notations and definitions.

Notation	Definition
F	Extracted current frame from the video stream.

F'	Extracted latest frame from the video stream.
fps	Frames per second.
H	Hue image of F' .
T_N	The current time.
T_L	The last time when it detected the chair.
O_i	The bounding box of the i^{th} object's detection result.
fid	The number of frames extracted from the video stream.
fid^N	The number of the last frame when the person was near the chair.
fid^P	The number of the last frame when it detected the person.
fid^C	The number of the frame when the person is obscured.
C_i	The bounding box of the i^{th} chair.
C_n	The chair that is the nearest chair to the i^{th} person.
P_i	The bounding box of the i^{th} person in the current frame.
\bar{P}_i	The bounding box of the i^{th} person's tracking result.
P_i^N	The latest bounding box of the i^{th} person when he is near the chair.
P_i^P	The latest bounding box of the i^{th} person when it detects the person.
P_i^S	The bounding box of the i^{th} person before 1.5 s.
$P_{i,fid}$	The bounding box of the i^{th} person in the fid^{th} frame.
$P_{i,fid}^H$	The hue images of the $P_{i,fid}$ area in H .
D_x	The horizontal distance from P to C .
D_y	The vertical distance from P to C .
D_c	The vertical distance from P .center to C .center.
B_i	The i^{th} binary back projection images of each $P_{i,fid}^H$.
M_i	Boolean, true while it detects the i^{th} person in the current frame.
$M_{i,fid}$	Boolean, true while it detects the i^{th} person in the fid^{th} frame.
S_i	The state of the i^{th} person in the current frame.
$S_{i,fid}$	The state of the i^{th} person in the fid^{th} frame.

For person tracking, we use Camshift to track the person and ensure that he/she still appears in the next frame. Algorithm 2 shows the steps of person tracking. First, we convert the latest frame F' from the Red, Green and Blue (RGB) color space to the Hue-Saturation-Value (HSV) color space and then extract the hue histogram in the HSV color space. The obtained image is called H , the hue histogram of F' . Next, we extract a specific hue histogram $P_{i,fid-1}^H$ in each area of $P_{i,fid-1}$ in H . We then use the Back Projection function in Open Source Computer Vision (OpenCV) [37] to obtain the back projection of each $P_{i,fid-1}^H$ in H ; then, we can find similar features in the image. To reduce the noise, we enhance the back projection images and the human reigns. For this, we convert the back projection images into a binary scale and perform the Erosion and Dilation OpenCV function. The obtained binary back projection images are called B_i . Then, we take $P_{i,fid-1}$ and B_i as input to perform Camshift, which is a widely used object tracking method. The main idea of mean shift is as follows: given a tracking window that contains a set of points, such as back projection images, mean shift will move the tracking window that contains the most points. The obtained bounding box of person tracking is \bar{P}_i . If the center of \bar{P}_i is in the bounding box of that person P_i , then we ensure that the person still appears in the frame. If it cannot find the person, then it executes fall detection.

Algorithm 1 Object detection

- Input: F
- Output: C_i , $P_{i,fid}$
1. F = Extract (video streaming)
 2. Fill the image size of F to a multiple of 32
 3. Load YOLOv3 model parameter
 4. $O_i \leftarrow$ YOLO detector (F)
 5. for all O_i do
 6. if $T_L = 0$ then
 7. if $O_i.class_name =$ Chair then

```

8.            $C_i \leftarrow O_i$ 
9.    else if  $O_i.\text{class\_name} = \text{Bench}$  then
10.       $C_i \leftarrow O_i$ 
11.       $C_i.\text{height} = 2 * C_i.\text{height}$ 
12.       $C_i.\text{top} = C_i.\text{top} + C_i.\text{height}$ 
13.    if  $O_i.\text{class\_name} = \text{Person}$  then
14.       $P_{i,fid} \leftarrow O_i$ 
15.    end for
16.     $T_L \leftarrow T_N$ 

```

Algorithm 2 Person tracking

Input: F' , $P_{i,fid}$
Output: P_i

```

1.     $H \leftarrow \text{GetHue}$  (Convert  $F'$  from RGB to HSV)
2.    for all  $P_{i,fid-1}$  do
3.       $P_{i,fid-1}^H \leftarrow \text{Get } P_{i,fid-1} \text{ area hue in } H$ 
4.    end for
5.     $B_i = \text{Back projection}(P_{i,fid-1}^H, H)$ 
6.     $B_i = \text{Convert to binary}(B_i)$ 
7.     $B_i = \text{Erosion}(B_i)$ 
8.     $B_i = \text{Dilation}(B_i)$ 
9.    for all  $P_{i,fid-1}$  do
10.       $\bar{P}_i \leftarrow \text{Camshift}(P_{i,fid-1}, B_i)$ 
11.       $M_{i,fid} = \text{false}$ 
12.      for all  $P_{i,fid-1}$  do
13.        if  $(\bar{P}_i.\text{center} \in P_i)$  then
14.           $M_{i,fid} = \text{true}$ 
15.          Break
16.        end for
17.        if  $(M_{i,fid} = \text{false})$  then
18.          Execute fall detection
19.    end for

```

4.3. Person Positioning and Posture Recognition

Algorithm 3 shows the steps of person positioning. For each person in the frame, we pair the person and the chair nearest to the person by computing the distance between the person and the chair. If the horizontal distance D_x from P to C is less than or equal to twice the width of the person's bounding box and the vertical distance D_y from P to C is less than or equal to half the height of the chair's bounding box, we consider that the person is near the chair. Otherwise, we consider that the person is far from the chair. Figure 2 shows the flow of the state when the person is far from the chair. If the person is far from the chair, IFADS detects falls while they are walking. In addition, if D_x is less than or equal to half the width of the person's bounding box, we consider that the person is beside the chair. Otherwise, we consider that the person is near the chair. When the person is beside the chair, since the elderly usually fall when they try to sit down on, or stand up from, a chair, IFADS executes posture recognition.

Algorithm 3 Person positioning

Input: C_i , P_i
Output: C_n, S_i

```

1.    for all  $P_i$  do

```

```

2.      for all  $C_i$  do
3.          if ( $j = 1$ ) then
4.               $C_n \leftarrow C_j$ 
5.          else
6.              if  $((P_i.X - C_j.X)^2 + (P_i.Y - C_j.Y)^2 \leq (P_i.X - C_n.X)^2 + (P_i.Y - C_n.Y)^2)$  then
7.                   $C_n \leftarrow C_j$ 
8.          end for
9.          if  $(D_x \leq 2 * P_i.W)$  and  $(D_y \leq 1/2 * C_n.H)$  then
10.              $S_i \leftarrow$  Near the chair
11.             if  $(D_x \leq 1/2 * P_i.W)$  then
12.                  $S_i \leftarrow$  Beside the chair
13.             Execute posture recognition
14.         else
15.              $S_i \leftarrow$  Far from the chair
16.             Execute fall detection
17.     end for

```

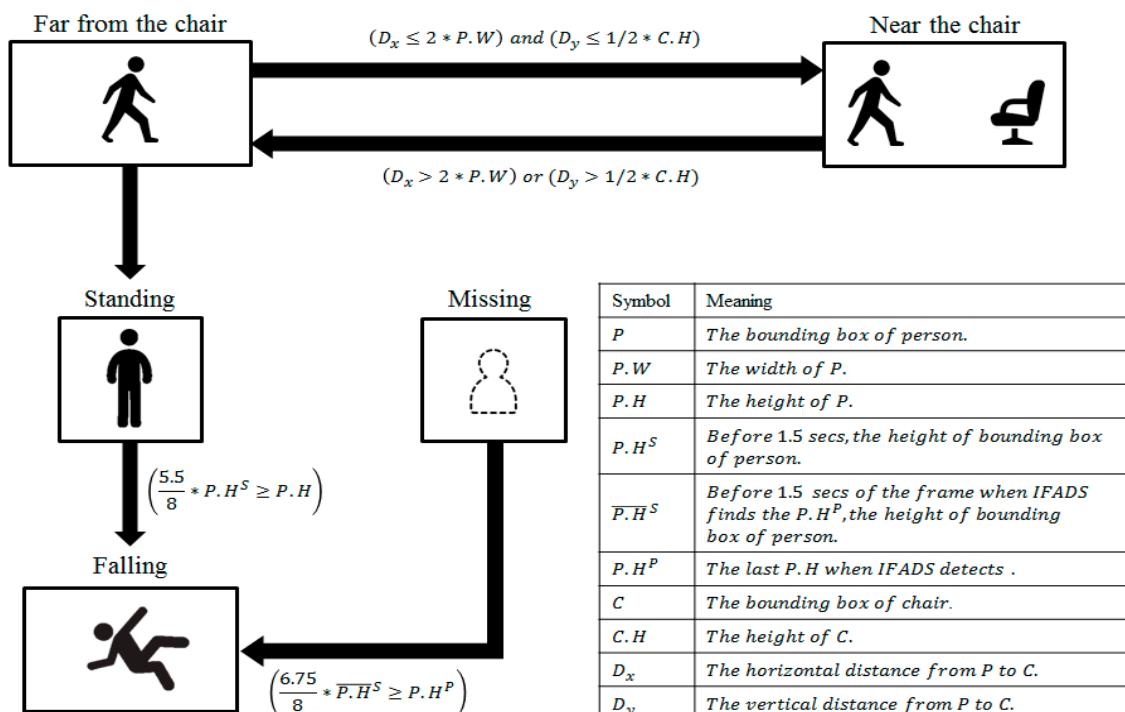


Figure 2. The flow of the state when the person is far from the chair.

Algorithm 4 shows the steps of the posture recognition pre-progress. When we execute posture recognition, we need the latest height of the person P_i^N when the person is near the chair. The height is the most objective height, since the person and the chair are on the same horizontal plane, instead of subject to the effect of the angle of the camera or the distance from the camera.

Algorithm 4 Posture recognition pre-progress

Input: $P_{i,fid}, S_{i,fid}, fid$

Output: P_i^N

```

1.      for  $fid^N = fid$  to 0 do
2.          if  $(S_{i,fid^N} =$  Near the chair) then
3.               $P_i^N \leftarrow P_{i,fid^N}$ 
4.          return  $P_i^N$ 

```

5. end for

Algorithm 5 shows the steps of posture recognition (between in progress and beside the chair). First, we execute posture recognition pre-progress to obtain the bounding box of person P_i^N when the person is near the chair. If the state of the person in the latest frame $S_{i,fid-1}$ is beside the chair, then IFADS detects the posture of the person by comparing the height of P_i^N and P_i . We adopt the idea of some specialists that the body height equals eight heads [34]. Since a person bending his/her leg is almost 7.5/8 times the height of the person, if the height of the person's bounding box in the current frame P_i is less than or equal to 7.5/8 times the height of the person's bounding box P_i^N , we consider that the person is not standing completely. Conversely, if the height of the person's bounding box in the current frame is more than 7.5/8 times the height of the person's bounding box P_i^N , then we consider that the person is standing completely. Figure 3 shows the flow of the state when the person is near or beside the chair.

Algorithm 5 Posture recognition (between in progress and beside the chair)

Input: P_i , $S_{i,fid}$, P_i^B

Output: S_i

1. Execute posture recognition pre-progress (fid)
 2. if ($S_{i,fid-1}$ = Beside the chair) then
 3. if ($P_i.height \leq 7.5/8 * P_i^N.height$) then
 4. $S_i \leftarrow$ In progress
 5. else if ($S_{i,fid-1}$ = In progress)
 6. if ($P_i.height > 7.5/8 * P_i^N.height$) then
 7. $S_i \leftarrow$ Beside the chair
-

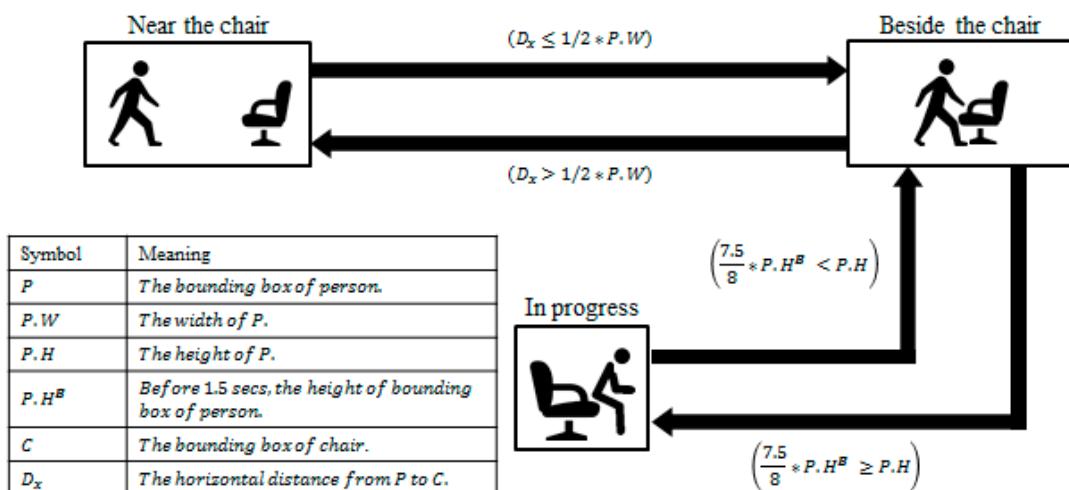


Figure 3. The flow of the state when the person is near or beside the chair.

Algorithm 6 shows the steps of posture recognition (in progress, sitting, and in danger). First, we execute posture recognition pre-progress to obtain the bounding box of person P_i^N when the person is near the chair. If the state of the person in the latest frame $S_{i,fid-1}$ is in progress, sitting, or in danger, we recognize the posture of the person. Since, when a person is sitting, the top of the person should be higher than the top of the chair, if the top of the person is lower than the top of the chair, we consider that the person is in danger and execute fall detection. Conversely, if the top of the person is higher than the top of the chair, since the center of C_n is the center of the faceplate of the chair, if the center of C_n is in P_i , it means that the person is in the process of sitting down on or standing up from the chair. On the other hand, since the center of P_i is the waist, if the center of P_i

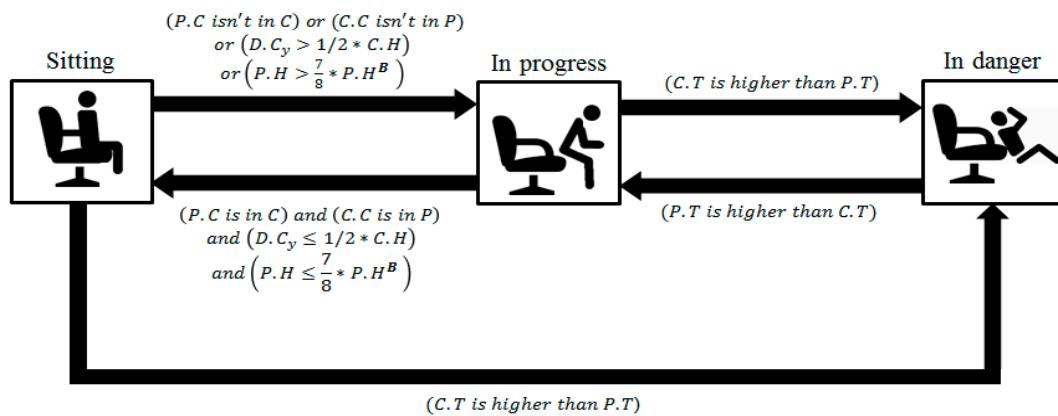
is in C_n , it means that the person is trying to sit on the chair. Even if the person falls, he/she can still sit on the chair. If the distance from the center of P_i to the center of C_n is more than half the height of C_n , it means that the person is not sitting on the chair. Finally, if the person is sitting correctly, the person should keep his/her back straight and bend their knees about 90 degrees. That height is about 7/8 times the height of the person when he/she is standing. If the height of the person's bounding box in the current frame P_i is less than or equal to 7/8 times the height of the person's bounding box P_i^B , then we consider that the person is sitting. Figure 4 shows the flow of the state when the person is sitting, in progress, or in danger. If the person is in danger or if IFADS cannot detect the person, it executes fall detection.

Algorithm 6 Posture recognition (between in progress, sitting, and in danger)

Input: P_i , $S_{i,fid}$, P_i^B

Output: S_i

1. Execute posture recognition pre-progress (fid)
 2. if ($S_{i,fid-1} = \text{In progress}$) or ($S_{i,fid-1} = \text{Sitting}$) or ($S_{i,fid-1} = \text{In danger}$) then
 3. if ($P_i.\text{top}$ higher than $C_n.\text{top}$) then
 4. if ($P_i.\text{center}$ in C_n) and ($C_n.\text{center}$ in P_i) then
 5. if ($D.C_y \leq 1/2 * C_n.\text{height}$) then
 6. if ($P_i.\text{height} \leq 7/8 * P_i^B.\text{height}$) then
 7. $S_i \leftarrow \text{Sitting}$
 8. else
 9. $S_i \leftarrow \text{In progress}$
 10. else
 11. $S_i \leftarrow \text{In progress}$
 12. else
 13. $S_i \leftarrow \text{In progress}$
 14. else
 15. $S_i \leftarrow \text{In danger}$
 16. Execute fall detection
-



Symbol	Meaning
P	The bounding box of person.
$P.C$	The center of P .
$P.T$	The top of P .
$P.H$	The height of P .
$P.H^B$	The last $P.H$ when the state is near the chair.

Symbol	Meaning
C	The bounding box of chair
$C.C$	The center of C .
$C.T$	The top of C .
$C.H$	The height of C .
$D.C_y$	The vertical distance from $P.C$ to $C.C$.

Figure 4. The flow of the state when the person is sitting, in progress, or in danger.

Since the person may be obscured by other objects, we recognize whether the person is obscured by other objects. Algorithm 7 shows the steps of posture recognition (obscured). The current surveillance camera can record at least five frames per second, and the person's height cannot change significantly within 0.2 s. Thus, if the height of the person is less than or equal to half the height in the last frame, we consider that the person is obscured. Conversely, if the state of the person is obscured in the last frame, but the height of P_i is higher than the height of $P_{i,fid}^{c-1}$ that is not obscured by other objects, we consider that the person is not obscured by objects.

Algorithm 7 Posture recognition (obscured)

Input: P_i , $S_{i,fid}$
 Output: S_i

1. if ($S_{i,fid-1}$!= Obscured) and ($S_{i,fid-1}$!= Falling) then
2. if ($P_i.height \leq 1/2 * P_{i,fid-1}.height$) then
3. $S_i \leftarrow$ Obscured
4. $fid^c \leftarrow fid$
5. else
6. if ($P_i.height \geq P_{i,fid}^{c-1}.height$) then
7. $S_i \leftarrow S_{i,fid}^{c-1}$

4.4. Fall Detection

Algorithm 8 shows the steps of fall detection. If IFADS can detect the person, we detect the fall by the previous state of the person. If the state of the person in the latest frame is standing, it means that the person falls while walking. We obtain the bounding box of the person P_i^S . Since a person usually falls within 1.5 s, the person's height 1.5 s before is the normal height of the person. If the person falls and is unable to stand up, the maximum of the person's height is similar to his/her height when he/she is doing a heel sit. Because a person bends his/her thigh and shank when he/she does a heel sit, the height of the person is about 6/8 times the height of the person when he/she is standing. Unlike when doing a heel sit, when a person falls, because he/she feels pain, the person bends his/her back instead of keeping his/her back straight, resulting in a difference of about one half of a head in height. As mentioned above, if the height of the person is less than or equal to 5.5/8 times the height 1.5 s before, we consider that the person falls. Conversely, if the state of the person in the latest frame is in danger, it means the person is beside the chair, trying to sit down on or stand up from the chair. For that, we obtain the bounding box of the person P_i^B when the person is near the chair by posture recognition pre-progress. If the height of the person is less than or equal to 5.5/8 times the height when the person is beside the chair, we likewise consider that the person falls. Conversely, if IFADS cannot detect the person, it executes fall detection for the missing person to detect a fall. Then, if the person remains in the fallen state for more than 3 s, IFADS triggers the alarm.

Algorithm 8 Fall detection

Input: $fid, P_i, P_{i,fid}, M_i, S_{i,fid}, P_i^B$
 Output: S_i

1. if (M_i = true) then
2. if ($S_{i,fid-1}$ = Standing) then
3. $P_i^S \leftarrow P_{i,fid-fps*1.5}$
4. if ($5.5/8 * P_i^S.height \geq P_i.height$) then
5. $S_i \leftarrow$ Falling
6. else if ($S_{i,fid-1}$ = In danger) then
7. Execute posture recognition pre-progress (fid)
8. if ($5.5/8 * P_i^B.height \geq P_i.height$) then
9. $S_i \leftarrow$ Falling

-
10. Else
 11. Execute fall detection for missing person
 12. if (S_i = Falling more than 3 s) then
 13. Trigger alarm
-

When IFADS cannot detect the person, it executes fall detection to detect a fall. Before fall detection, we need to obtain the latest bounding box of the missing person in the previous frame. Algorithm 9 shows the steps of fall detection for a missing person pre-progress. From the current frame to the first frame, we find the latest frame when IFADS detects the person. If IFADS finds the person, it returns the person's bounding box P_i^P and the number of that frame fid^N .

Algorithm 9 Fall detection for a missing person pre-progress

- Input: fid , $M_{i,fid}$
 Output: P_i^P , fid^P
1. for $fid^N = fid$ to 0 do
 2. if $M_{i,fid} = \text{true}$ then
 3. $P_i^P \leftarrow P_{i,fid^P}$
 4. return P_i^P , fid^P
 5. end for
-

Algorithm 10 shows the steps of fall detection for a missing person. First, we obtain the bounding box of the person P_i^P from when IFADS detected the missing person the last time to evaluate whether the person fell. If the state of the person is far from the chair or near the chair, it means that the person may have fallen while walking. If the person cannot be detected, we cannot obtain the height of the person's bounding box. Thus, if the person's height in the fid^P frame is less than or equal to the average of the height of standing and falling, we consider that the person may have fallen. As mentioned above, if the height of the person in the fid^P frame is less than or equal to $6.75/8$ times the height 1.5 s before the fid^P frame, we consider that the person may have fallen. Conversely, if the state of the person in the fid^P frame is beside the chair, sitting, in progress, or in danger, it means that the person may have fallen while sitting down on or standing up from the chair. We likewise obtain the height of the person when he/she is near the chair by posture recognition pre-progress. Similarly, if the height of P_i^P is less than or equal to $6.75/8$ (the average of 1 and $5.5/8$) times the height when he/she is near the chair, we consider that the person may have fallen. If the state of the person in fid^P is falling, we consider that the person remains in the state of falling. Figure 5 shows the flow of fall detection when the person's state is in danger or missing.

Algorithm 10 Fall detection for missing person

- Input: fid , $P_{i,fid}$, $S_{i,fid}$
 Output: S_i
1. Execute fall detection for missing person pre-progress (fid)
 2. if (S_{i,fid^N} = Far from the chair or Near the chair) then
 3. $P_i^S \leftarrow P_{i,fid^P-fps*1.5}$
 4. if ($(6.75/8 * P_i^S.height \geq P_i^P.height)$)
 5. $S_i \leftarrow \text{Falling}$
 6. else if (S_{i,fid^N} = Beside the chair or Sitting or In progress or In danger) then
 7. Execute posture recognition pre-progress (fid^P)
 8. if ($(6.75/8 * P_i^B.height \geq P_i^P.height)$)
 9. $S_i \leftarrow \text{Falling}$
 10. else if (S_{i,fid^N} = Falling)
 11. $S_i \leftarrow \text{Falling}$
-

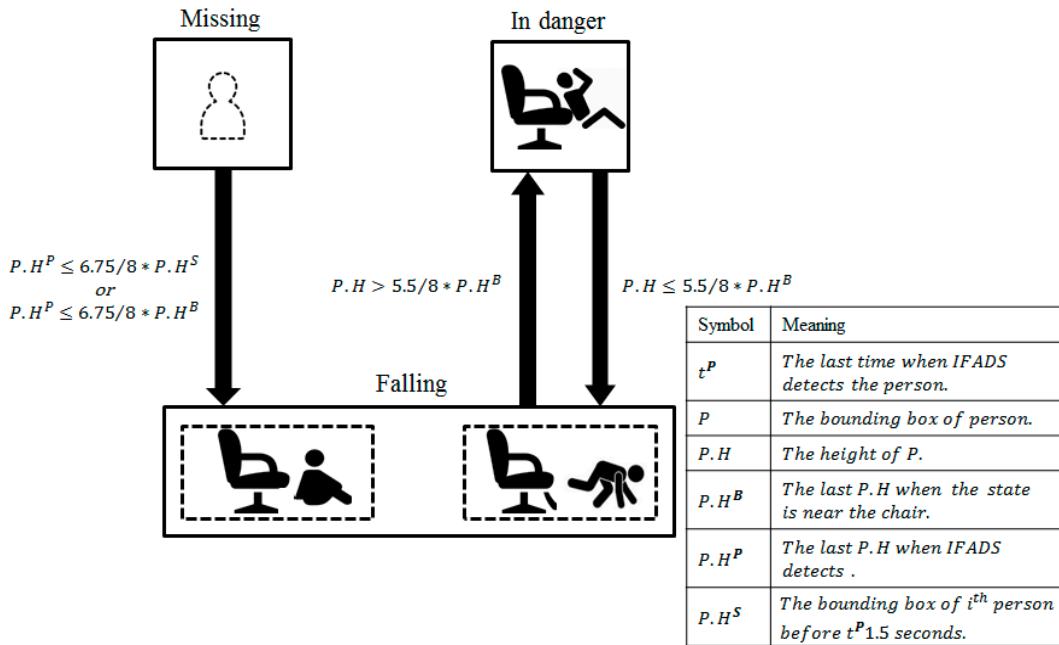


Figure 5. The flow of fall detection when the person's state is in danger or missing.

5. Experiment

5.1. Experimental Setup

In order to investigate the effectiveness of IFADS, we not only simulated different fall scenarios, but also adopted YouTube and Giphy videos that captured real falls. In the simulated fall scenarios and the falls captured in YouTube and Giphy videos, we conducted experiments with different environments, rotation angles of the fall, and perspectives to verify the accuracy of IFADS. In addition, the falls captured in YouTube and Giphy videos were multiple-person and multiple-chair scenarios to prove that IFADS can be used in existing surveillance systems.

5.2. Test Cases: Common Situation

For the common situation, there are three tests. There are 23 cases in each test, including 9 non-fall cases and 14 fall cases. Case 1 to Case 4 in the non-fall cases are walking cases to verify that IFADS does not detect a fall while the tester is walking. Figure 6 shows the illustration of the walking cases. Case 1 and Case 2 both show that the tester walks across behind the chair, and the tester is far from the chair in Case 1 and near the chair in Case 2. Case 3 and Case 4 are both situations where the tester walks across in front of the chair, and the tester is far from the chair in Case 3 and near the chair in Case 4. On the other hand, Case 5 to Case 9 in the non-fall cases are sitting cases to verify that IFADS does not detect a fall while the tester is sitting with any rotation angle. Figure 7 shows the illustration of the sitting cases. Cases 5, 6, and 7 are the situations where the tester faces the chair, walks to the chair, sits, and then stands up and walks away. The difference between these cases is that the tester has his/her back to the chair, faces the camera, and turns his/her back to the camera while sitting in Cases 5, 6, and 7, respectively. Case 8 is the situation where the tester walks to the chair with his/her back to the camera, sits, and then stands up and walks away. Case 9 is the situation where the tester walks to the chair while facing the camera, sits, and then stands up and walks away. From Case 10 to Case 23, there are 14 fall cases. Figure 8 shows the illustration of the fall cases. From Case 10 to Case 16, the tester falls when he/she tries to sit on the chair, and from Case 17 to Case 23, the tester falls when he/she tries to stand up from the chair. The difference between these cases is that the rotation angle of the fall is different. Figure 9 shows the different rotation angles of the fall. The purpose of these fall cases is to verify that IFADS can detect falls with different rotation angles and different postures.

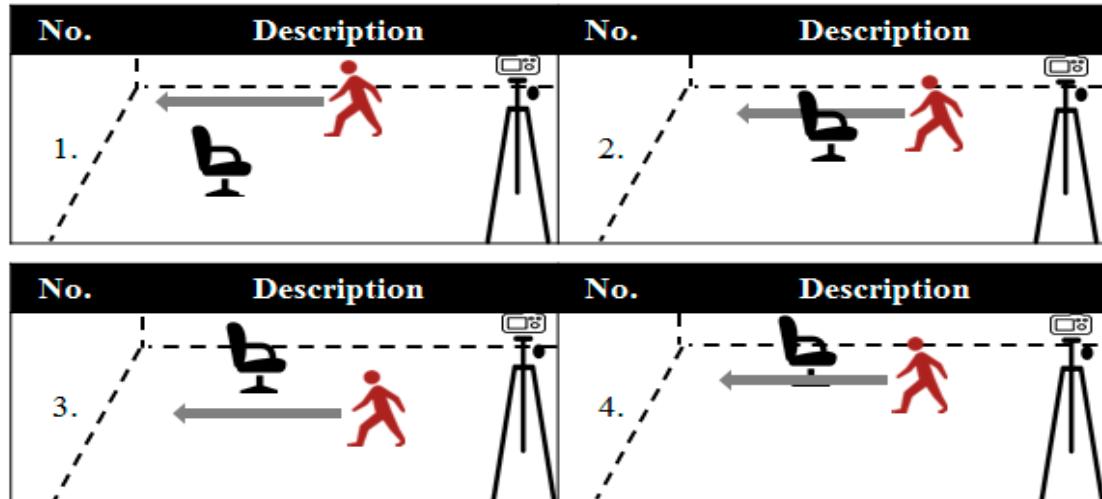


Figure 6. An illustration of the walking cases.

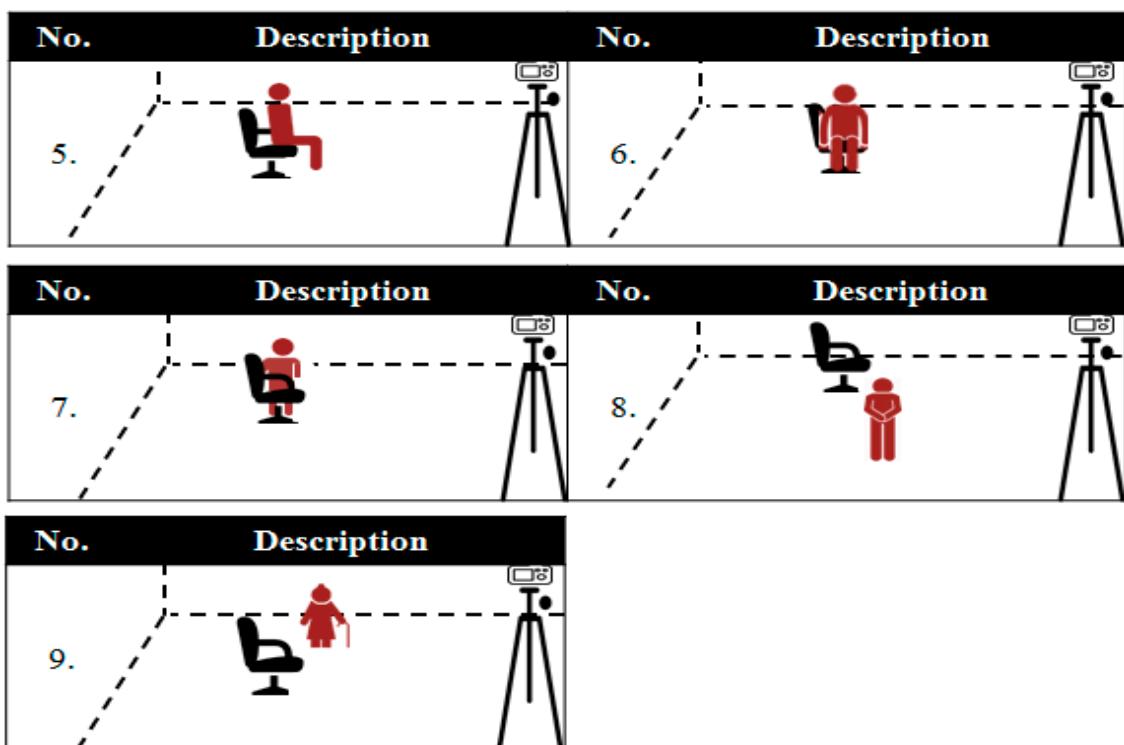


Figure 7. An illustration of the sitting cases.

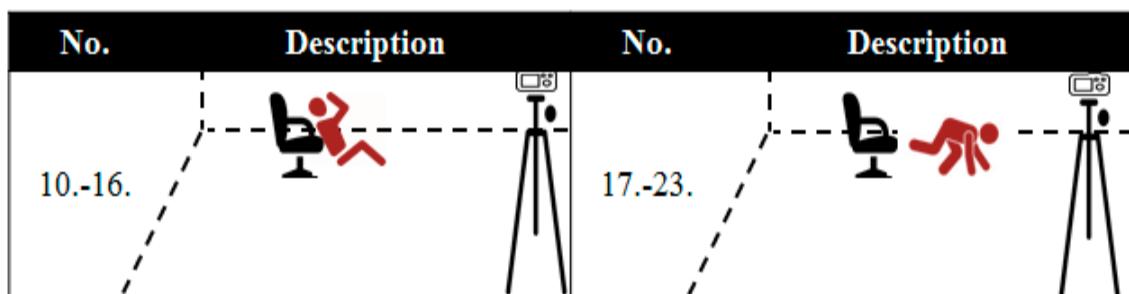


Figure 8. An illustration of the cases with a fall.

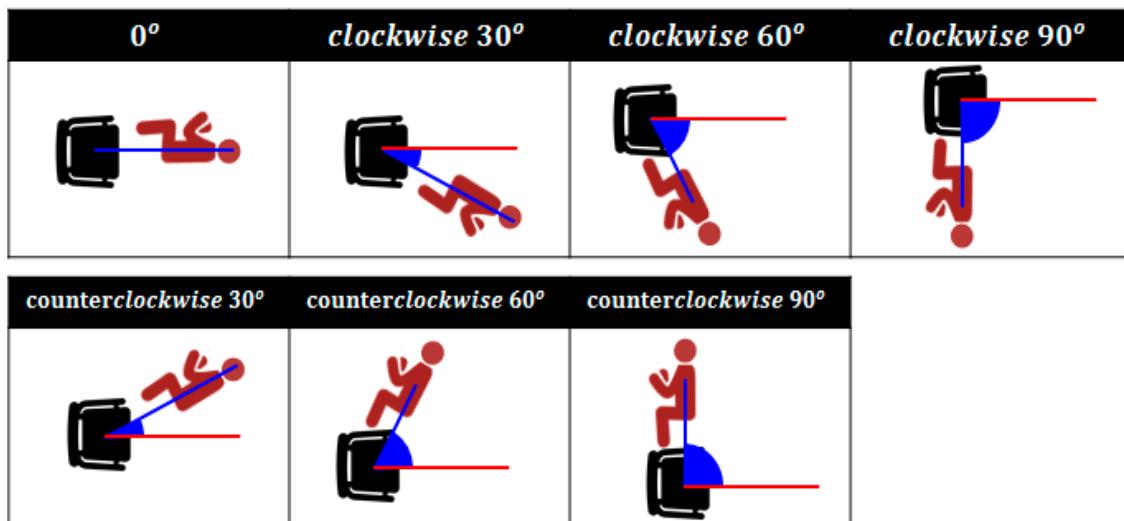
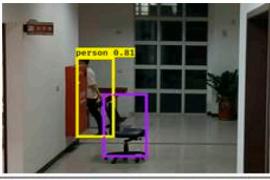
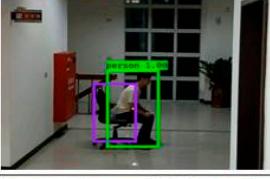
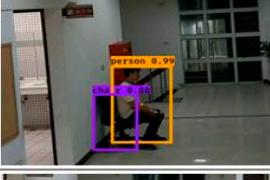


Figure 9. Different rotation angles of falls.

For the experiment, first, we mark the chair purple. As Table 2 shows, we mark the testers in different colors to distinguish between the different postures of the testers. Figure 10 shows the remaining results of the common situation. Since the tester usually walks across (in front of or behind) the chair instead of sitting on the chair when he/she finds that there is a chair, from Case 1 to Case 4 in Figure 10, we test when the tester walks across (in front of or behind) the chair and does not sit down from a different perspective. In addition, the tester walks a different distance from the tester to the chair and takes a different route. As the tester's height does not change significantly within 1.5 s, IFADS does not detect a fall. From Case 5 to Case 9, we test when the tester walks, sits on the chair, stands up from the chair, and leaves with a different rotation angle of sitting. Since the top of the tester remains higher than the top of the chair, IFADS does not detect a fall. From Case 5 to Case 8, IFADS accurately recognizes the posture of the tester as sitting. IFADS recognizes the posture of the tester as in progress in Case 9, because the tester does not sit at the center of the chair's faceplate. However, if an elderly person sits at the end of a chair, he/she may slide out from the chair easily. Then, we test when the tester falls off the chair while sitting with a different rotation angle of the fall from Case 10 to Case 16, and test when the tester falls off the chair while standing up from the chair in other cases. IFADS detects the fall accurately in these cases. In Cases 10, 21, 22, and 23, IFADS cannot detect the tester, because the falling tester's human features disappear from the frame, but IFADS can still detect the fall because of the state tracking.

Table 2. List of colors and definitions.

Color.	Definition
Blue	The person's state is far from the chair.
Sky blue	The person's state is near the chair.
Yellow	The person's state is beside the chair.
Orange	The person's state is in progress.
Green	The person's state is sitting on the chair.
Red	The person's state is falling.

Case	Output	Accuracy	Case	Output	Accuracy
Walking (#1)		Correct	Walking (#2)		Correct
Walking (#3)		Correct	Walking (#4)		Correct
Sitting (#5)		Correct	Sitting (#6)		Correct
Sitting (#7)		Correct	Sitting (#8)		Correct
Case	Output	Accuracy	Case	Output	Accuracy
Sitting (#9)		Correct	Falling (#10)		Correct
Falling (#11)		Correct	Falling (#12)		Correct
Falling (#13)		Correct	Falling (#14)		Correct
Falling (#15)		Correct	Falling (#16)		Correct

Case	Output	Accuracy	Case	Output	Accuracy
Falling (#17)		Correct	Falling (#18)		Correct
Falling (#19)		Correct	Falling (#20)		Correct
Falling (#21)		Correct	Falling (#22)		Correct
Falling (#23)		Correct			

Figure 10. The results of the common situation test cases.

Among these cases, the color of the tester's clothes may cause a fault in IFADS detection, because if the color of the clothes is similar to the color of the chair, the height of the tester that IFADS detects may be wrong. On the other hand, the height ratio of the tester to the chair may cause a fault in IFADS detection, because if the person falls, the top of the tester may be higher than the chair. For that, we test the effect of the colors and the effect of the person–chair ratio on the accuracy.

Figure 11 shows the test results of the effect of the colors on the accuracy. In all test cases, IFADS detects correctly. However, in Cases 12, 13, 15, 16, 19, and 23 in Figure 11, the tester cannot be detected. In Cases 12, 15, and 19, the tester cannot be detected due to having a falling posture. In Case 13, the color of the tester's clothes is similar to the color of the chair, so IFADS cannot detect the tester. In Cases 16 and 23, IFADS cannot detect the tester or the tester's posture, and it is too difficult to find the human features, as the tester is obscured by the chair. In Case 20, the height of the tester's bounding box is not detected properly, since the colors of the tester's clothes and hair are similar to the color of the chair. Even if the height is not detected properly, IFADS can detect the fall, because the top of the tester's bounding box is lower than the top of the chair, and the height is the same as the fall while the tester is sitting on the floor.

Case	Output	Accuracy	Case	Output	Accuracy
Walking (#1)		Correct	Walking (#2)		Correct
Walking (#3)		Correct	Walking (#4)		Correct
Sitting (#5)		Correct	Sitting (#6)		Correct
Sitting (#7)		Correct	Sitting (#8)		Correct
Case	Output	Accuracy	Case	Output	Accuracy
Sitting (#9)		Correct	Falling (#10)		Correct
Falling (#11)		Correct	Falling (#12)		Correct
Falling (#13)		Correct	Falling (#14)		Correct
Falling (#15)		Correct	Falling (#16)		Correct

Case	Output	Accuracy	Case	Output	Accuracy
Falling (#17)		Correct	Falling (#18)		Correct
Falling (#19)		Correct	Falling (#20)		Correct
Falling (#21)		Correct	Falling (#22)		Correct
Falling (#23)		Correct			

Figure 11. The test results of the effect of colors on IFADS's accuracy.

Figure 12 shows the test results of the effect of the person–chair ratio on the accuracy. For all tests, IFADS detects correctly. The tester in Figure 12 is 0.15 m higher than the tester in Figure 10, and the chair in Figure 12 is 0.1 m lower than the chair in Figure 10. As mentioned above, the height ratio of the tester to the chair for Figure 10 is 1.74, and the height ratio of the tester to the chair for Figure 12 is 2.12. In all test cases, since IFADS detects the change in the person's state, it can detect correctly even if the tester cannot be detected, because the human features of the falling tester cannot be found in Cases 14, 15, 16, 17, 20, 21, 22, and 23. In Cases 14, 15, 16, 21, 22, and 23, IFADS cannot detect the testers, because some of the testers' features are obscured by the chair. Moreover, the falling posture cannot be detected in Case 17. In Cases 20 and 23, IFADS cannot detect the tester, because the tester's posture is too difficult to detect and the tester's human features are obscured by the chair. In all fall cases, IFADS can detect the fall even if the person is higher than twice the height of the chair.

Case	Output	Accuracy	Case	Output	Accuracy
Walking (#1)		Correct	Walking (#2)		Correct
Walking (#3)		Correct	Walking (#4)		Correct
Sitting (#5)		Correct	Sitting (#6)		Correct
Sitting (#7)		Correct	Sitting (#8)		Correct
Case	Output	Accuracy	Case	Output	Accuracy
Sitting (#9)		Correct	Falling (#10)		Correct
Falling (#11)		Correct	Falling (#12)		Correct
Falling (#13)		Correct	Falling (#14)		Correct
Falling (#15)		Correct	Falling (#16)		Correct

Case	Output	Accuracy	Case	Output	Accuracy
Falling (#17)		Correct	Falling (#18)		Correct
Falling (#19)		Correct	Falling (#20)		Correct
Falling (#21)		Correct	Falling (#22)		Correct
Falling (#23)		Correct			

Figure 12. The test results of the effect of the person–chair ratio on IFADS’s accuracy.

5.3. Test Cases: Other Situations

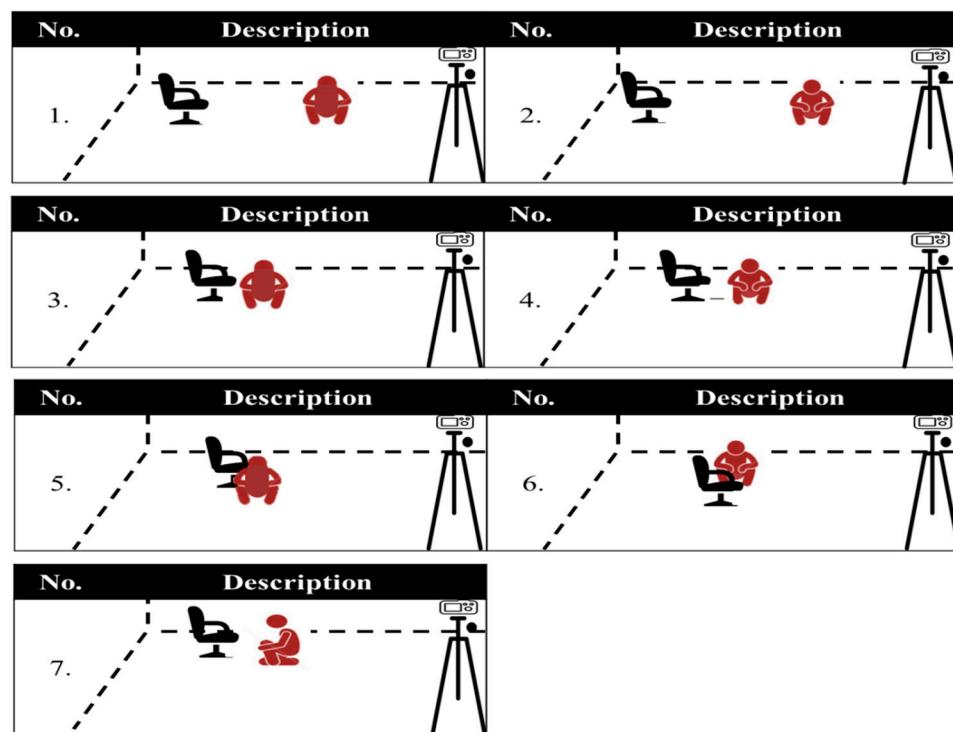
For the other situations, there are three tests: sitting on a bench with no seatback, squatting, and a high-angle shot. Apart from when lying on the floor, the top of the person is still higher than the top of the bench with no seatback. For that, if IFADS detects the object as the bench, then it considers the height of the bench as twice that. Then, we test when the tester is sitting on the bench with no seatback, and falls off the bench with no seatback, while sitting on the bench and standing up from the bench. Figure 13 shows the results of the bench with no seatback. In Case 1, the tester walks to the bench, sits down, and walks away. In Case 2, the tester walks to the bench, sits down, and falls while sitting down. In Case 3, the tester walks to the bench, sits, and falls while standing up from the bench. The results in Figure 13 show that IFADS detects correctly even with a bench with no seatback.

Case	Output	Accuracy	Case	Output	Accuracy
Sitting (#1)		Correct	Falling (#2)		Correct
Falling (#3)		Correct			

Figure 13. The results of the bench with no seatback test cases.

Squatting may cause a fault in IFADS detection. Figure 14 shows the illustration of the squatting cases. We test when the tester squats when far from the chair and when beside the chair for more than 3 s. Case 1 and Case 2 both show that the tester squats when far from the chair; the tester has his/her back to the camera in Case 1, and faces the camera in Case 2. Case 3 to Case 7 all show that the tester squats. Both Case 3 and Case 4 show that the tester squats when beside the chair; the tester has his/her back to the camera in Case 3, and faces the camera in Case 4. The tester squats in front of the chair, behind the chair, and while facing the chair in Cases 5, 6, and 7, respectively. Figure 15 shows the results of the squatting test cases. In Case 1 and Case 2, a fall is detected, since the height of the tester is less than 5.5/8 times the tester's height 1.5 s before and the tester maintains the state for more than 3 s. In the other cases in Figure 15, the tester squats when beside the chair. The tester is detected as a falling person in Case 3 and Case 7, since he bends his back more. The tester is not detected as a falling person in the other cases, since he keeps his back straight more. As mentioned, squatting may cause a fault in IFADS detection. However, it is dangerous for the elderly to squat for more than 3 s, since the elderly face muscle loss.

For a high-angle shot, the tester may cause a fault in IFADS detection, since the height of the person that IFADS detects may be wrong. Figure 16 shows the illustration of the difference between the common situation and a high-angle shot. We test when the tester walks across (in front of or behind) the chair, sits on the chair, and falls. Figure 17 shows the test results of the high-angle shot. IFADS detects the fall correctly, because it detects the fall by the change of the person's height and state.

**Figure 14.** An illustration of the squatting cases.

Case	Output	Accuracy	Case	Output	Accuracy
#1		Incorrect	#2		Incorrect
#3		Incorrect	#4		Correct
#5		Correct	#6		Correct
#7		Incorrect			

Figure 15. The results of the squatting test cases.

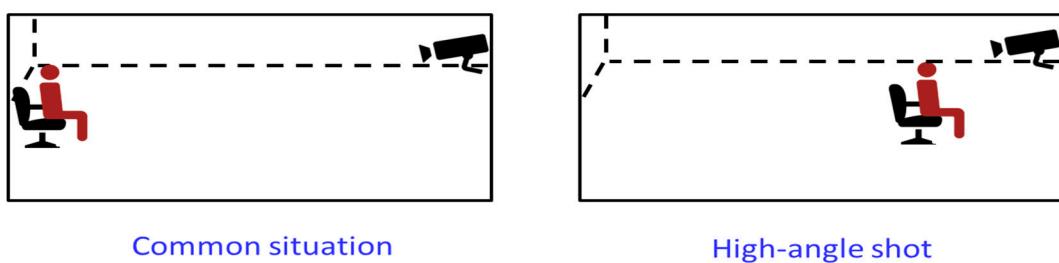


Figure 16. An illustration of a common situation and a high-angle shot.

Case	Output	Accuracy	Case	Output	Accuracy
Walking		Correct	Sitting		Correct
Falling (try to sit)		Correct	Falling (try to stand up)		Correct

Figure 17. The results of the high-angle shot test case.

5.4. Case Study

In order to further investigate the effectiveness of IFADS, we conducted 16 case studies of videos from YouTube and Giphy that captured real falls.

Figure 18 shows the results of the case studies for when a person falls while sitting. From Case 8 to Case 16, the person falls while walking. Case 1 [38] shows a man who walks to a chair and sits on the chair. However, when he sits and relies on the seatback, the seatback is broken, so he falls and his human features are lost from the frame while he is falling. Case 2 [39] shows that when an elderly man tries to sit on a wheelchair, the wheelchair is moving, so the elderly man falls. Since the elderly man is obscured by the bed, the elderly man cannot be detected. Case 3 to Case 7 [40] are multi-person and multi-chair situations. In Case 3, a man tries to sit but falls from a chair, and his body is obscured by other chairs. In Cases 4 and 5, a man falls from a chair. In Case 6, a man falls from a chair and is obscured by a table and another person. In Case 7, a man falls from a chair in a library.

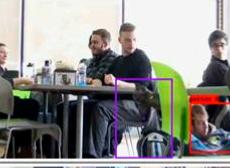
Case	Output	Accuracy	Case	Output	Accuracy
#1		Correct	#2		Correct
#3		Correct	#4		Correct
#5		Correct	#6		Correct
#7		Correct			

Figure 18. The results of the video case studies of falls while sitting.

Figure 19 shows the results of the case studies for when a person falls while walking. In Case 1 [41], a woman falls over on the street. In Case 2 [42], a man slips on the road when the road is wet. In Case 3 and Case 4 [42], an adult and a child slip on the ice. In Case 5 [43], an elderly man tries to use the walking aid and falls down. In Case 6 [44], an old man falls when his pet dog runs around him and trips him with a leash. In Case 7 [45], an elderly man falls because he feels dizzy. In Case 8 [46], shows an old woman runs along a corridor and hits an object, which makes her fall. In Case 9 [47], a man falls over while the floor is wet. In summary, IFADS detects the falls from all of these case study videos. In these cases, IFADS is proven to be a practical solution to help caregivers quickly identify falling people in public areas.

Case	Output	Accuracy	Case	Output	Accuracy
#1		Correct	#2		Correct
#3		Correct	#4		Correct
#5		Correct	#6		Correct
#7		Correct	#8		Correct
#9		Correct			

Figure 19. The results of the video case studies.

5.5. Performance Comparison

We compare IFADS with the method that Diraco et al. [6] has proposed, which is the most intuitive method. The method detects falls by the distance of the human centroid from the floor plane, and the authors consider that 0.4 m provides the best choice reduce false alarms. Thus, we recognize falls when the distance of the tester's centroid from the floor plane is less than 0.4 m manually. We test the method without using the video case studies, since we cannot know the real height of the people in the videos. Table 3 shows the results of that method. A true positive (TP) outcome occurs when the method correctly detects a fall case. A false positive (FP) outcome occurs when the method creates a false alarm. A false negative (FN) outcome occurs when the method misses a fall case. A true negative (TN) outcome occurs when the method correctly detects a non-fall case. When the tester falls while sitting down or trying to stand up, as in Case 10 to Case 16 in the common situation, and when the tester keeps his back straight or does not look down, it is difficult for the method to detect a fall. If the tester is taller than 180 cm, it is more difficult to detect a fall. On the other hand, if the person falls while standing up from a chair, as in Case 17 to Case 23 in the common situation, the method is likely to detect the fall.

Table 3. The results from the method [6].

	Actual: Fall	Actual: Non-Fall	
Predicted: Fall	27 (TP)	0 (FP)	Precision = 27/27 = 100%
Predicted: Non-fall	19 (FN)	37 (TN)	-
	Recall = 27/46 = 58.70%	-	Accuracy = 64/83 = 77.11%

TP, true positive; FP, false positive; FN, false negative; TN, true negative.

In addition, as the existing image-based method detects falls by recognizing the posture of the person in the current frame, we compare IFADS with the machine learning method. We consider that the method cannot detect a fall when the method cannot detect the person. As Figure 20 shows, the person may be unable to be detected or may be detected as another object. Thus, we adopted the TensorFlow object detection application programming interface (API) [48], which contains various models and neural networks (e.g., single shot detector (SSD) [49], MobileNet [50], region-based convolutional neural network (faster_RCNN) [51], inception [52], region-based fully convolutional networks (RFCN) [53], resnet101 (deep residual network) [54], and neural architecture search (NAS) [55]) to detect objects. IFADS adopts YOLO to detect falls. Table 4 shows the results of a falling person detected by the machine learning methods. In the fall cases, YOLO and ssd_mobilenet_v1_coco are faster than the other models, but the accuracy is only up to 59.7%. The accuracy of faster_RCNN_inception_v2_coco, RFCN_resnet101_coco, and faster_RCNN_NAS is up to 80.6%; however, they spend at least 10.03 s per frame, so they should not be adopted in real time. In summary, as Table 3 shows, the recall of the method that was proposed by Diraco et al. [6] was only 58.70%. Their method did not perform well in detecting falls while sitting down due to the poor prediction. Also, their method cannot identify the relationship between the person and environmental objects and may lead to an inaccurate result. Further, multiple cameras are required to determine the status of a walking person. As Table 4 shows, although YOLO and ssd_mobilenet_v1_coco can complete the computation quickly (0.13 s and 6.22 s, respectively), the accuracy was very low. Other methods, such as faster_RCNN_inception_v2_coco, RFCN_resnet101_coco, and faster_RCNN_NAS, can achieve a higher accuracy but require a longer computation time. Compared with the above-mentioned methods, IFADS performed better in detecting falls that occur while forward walking, sitting down, and standing up. IFADS can be used by not only nursing homes but also hospitals to improve the quality of health care facilities.

Table 4. The results of a falling person detected by the machine learning methods.

Model Name	Time	Success Cases	Fall Cases	Accuracy
IFADS	0.13 s	62	62	100%
YOLO	0.12 s	37	62	59.7%
ssd_mobilenet_v1_coco	6.22 s	25	62	40.3%

faster_RCNN_inception_v2_coco	10.03 s	44	62	71.0%
RFcn_resnet101_coco	18.73 s	50	62	80.6%
faster_RCNN_NAS	26.01 s	47	62	75.8%

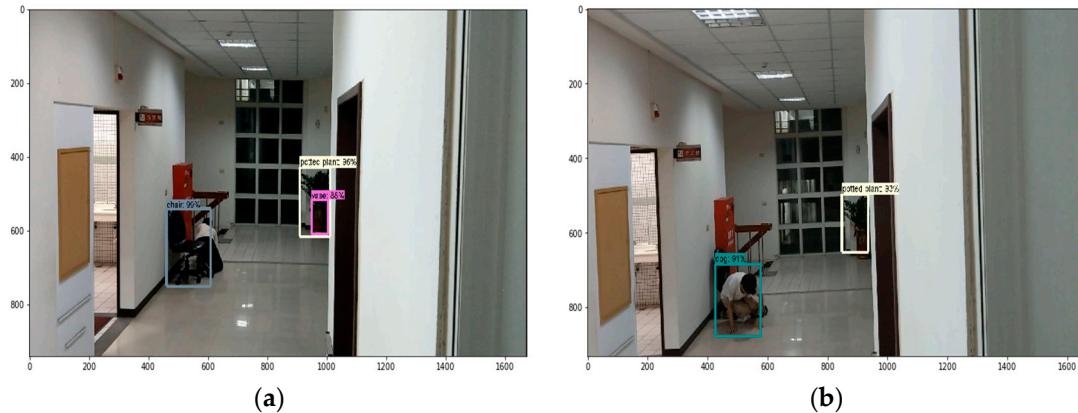


Figure 20. (a) The person is unable to be detected; (b) The person is detected as another object.

In conclusion, we analyzed the accuracy of IFADS with a total of 99 videos (83 test videos and 16 real videos).

Table 5 shows that, although IFADS creates four false alarms, it does not miss any fall cases. The four false alarms occurred because the tester squatted for more than 3 s; however, it is dangerous for the elderly to squat for more than 3 s. The precision was found to be 93.94%, and the recall was found to be 100%. In conclusion, the accuracy of IFADS is 95.96%. We have proven IFADS to be a practical solution to help caregivers, security staff, and responders quickly detect falls.

Table 5. Experimental results.

	Actual: Fall	Actual: Non-Fall	
Predicted: Fall	62 (TP)	4 (FP)	Precision = 62/66 = 93.94%
Predicted: Non-fall	0 (FN)	33 (TN)	-
	Recall = 62/62 = 100%	-	Accuracy = 95/99 = 95.96%
	Actual: Fall	Actual: Non-fall	
Predicted: Fall	62 (TP)	4 (FP)	Precision = 62/66 = 93.94%
Predicted: Non-fall	0 (FN)	33 (TN)	-
	Recall = 62/62 = 100%	-	Accuracy = 95/99 = 95.96%

6. Conclusions

In this paper, we presented the IFADS method to detect falls in the elderly using videos from surveillance systems or webcams. Given a video, IFADS can track a person, position the person based on the relationship between the person and the chair, recognize the person's posture by the change in the person's state and his/her position, and finally determine whether the person falls. We used 99 videos to test IFADS, and its accuracy was 95.96%. IFADS can be easily integrated into existing surveillance systems or webcams to help caregivers and security staff quickly detect falls. As a result, IFADS can be used by nursing homes to improve the quality of residential care facilities. In the future, we plan to extend IFADS to other scenarios, such as falling while picking something up from the ground and falling while getting out of bed.

Author Contributions: K.-L.L. contributed to the system's design and implementation, the experimental work, and the manuscript's drafting. E.T.-H.C. contributed to the system's design, the experiment's design, and the revision of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology (MOST) of Taiwan under Grant MOST 107-2628-E-224-001-MY3.

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

References

1. World Health Organization (WHO). WHO Falls—Fact Sheet. Available online: <http://www.who.int/mediacentre/factsheets/fs344/en/> (accessed on 5 September 2018).
2. Lai, C.; Chang, S.; Chao, H.; Huang, Y. Detection of Cognitive Injured Body Region Using Multiple Triaxial Accelerometers for Elderly Falling. *IEEE Sens. J.* **2011**, *11*, 763–770, doi:10.1109/JSEN.2010.2062501.
3. Ando, B.; Baglio, S.; Lombardo, C.O.; Marletta, V. A Multisensor Data-Fusion Approach for ADL and Fall classification. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1960–1967, doi:10.1109/tim.2016.2552678.
4. Feng, G.; Mai, J.; Ban, Z.; Guo, X.; Wang, G. Floor Pressure Imaging for Fall Detection with Fiber-Optic Sensors. *IEEE Pervasive Comput.* **2016**, *15*, 40–47, doi:10.1109/MPRV.2016.27.
5. Auvinet, E.; Multon, F.; Saint-Arnaud, A.; Rousseau, J.; Meunier, J. Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 290–300, doi:10.1109/TITB.2010.2087385.
6. Diraco, G.; Leone, A.; Siciliano, P. An active vision system for fall detection and posture recognition in elderly healthcare. In Proceedings of the 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), Dresden, Germany, 8–12 March 2010; pp. 1536–1541.
7. Brulin, D.; Beneteth, Y.; Courtial, E. Posture recognition based on fuzzy logic for home monitoring of the elderly. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 974–982, doi:10.1109/TITB.2012.2208757.
8. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, USA, 8–14 December 2001; p. I.
9. Robinovitch, S.N.; Feldman, F.; Yang, Y.; Schonnop, R.; Lueng, P.M.; Sarraf, T.; Sims-Gould, J.; Loughin, M. Video capture of the circumstances of falls in elderly people residing in long-term care: An observational study. *Lancet* **2013**, *381*, 47–54, doi:10.1016/S0140-6736(12)61263-X.
10. Rucco, R.; Sorriso, A.; Liparoti, M.; Ferraioli, G.; Sorrentino, P.; Ambrosanio, M.; Baselice, F. Type and Location of Wearable Sensors for Monitoring Falls during Static and Dynamic Tasks in Healthy Elderly: A Review. *Sensors* **2018**, *18*, 1613, doi:10.3390/s18051613.
11. Tong, L.; Song, Q.; Ge, Y.; Liu, M. HMM-Based Human Fall Detection and Prediction Method Using Tri-Axial Accelerometer. *IEEE Sens. J.* **2013**, *13*, 1849–1856, doi:10.1109/JSEN.2013.2245231.
12. Abeyruwan, S.W.; Sarkar, D.; Sikder, F.; Visser, U. Semi-Automatic Extraction of Training Examples from Sensor Readings for Fall Detection and Posture Monitoring. *IEEE Sens. J.* **2016**, *16*, 5406–5415, doi:10.1109/JSEN.2016.2559804.
13. Pannurat, N.; Thiemjarus, S.; Nantajeewarawat, E. A Hybrid Temporal Reasoning Framework for Fall Monitoring. *IEEE Sens. J.* **2017**, *17*, 1749–1759, doi:10.1109/jsen.2017.2649542.
14. Liu, J.; Lockhart, T.E. Development and evaluation of a prior-to-impact fall event detection algorithm. *IEEE Trans. Biomed. Eng.* **2014**, *61*, 2135–2140, doi:10.1109/TBME.2014.2315784.
15. Lustrek, M.; Gjoreski, H.; Vega, N.G.; Kozina, S.; Cvetkovic, B.; Mirchevska, V.; Gams, M. Fall Detection Using Location Sensors and Accelerometers. *IEEE Pervasive Comput.* **2015**, *14*, 72–79, doi:10.1109/MPRV.2015.84.
16. Pierleoni, P.; Belli, A.; Maurizi, L.; Palma, L.; Pernini, L.; Paniccia, M.; Valenti, S. A Wearable Fall Detector for Elderly People Based on AHRS and Barometric Sensor. *IEEE Sens. J.* **2016**, *16*, 6733–6744, doi:10.1109/jsen.2016.2585667.
17. Sabatini, A.M.; Ligorio, G.; Mannini, A.; Genovese, V.; Pinna, L. Prior-to- and Post-Impact Fall Detection Using Inertial and Barometric Altimeter Measurements. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2016**, *24*, 774–783, doi:10.1109/TNSRE.2015.2460373.
18. Ejupi, A.; Brodie, M.; Lord, S.R.; Annegarn, J.; Redmond, S.J.; Delbaere, K. Wavelet-Based Sit-To-Stand Detection and Assessment of Fall Risk in Older People Using a Wearable Pendant Device. *IEEE Trans. Biomed. Eng.* **2017**, *64*, 1602–1607, doi:10.1109/TBME.2016.2614230.
19. Daher, M.; Diab, A.; El Badaoui El Najjar, M.; Ali Khalil, M.; Charpillet, F. Elder Tracking and Fall Detection System Using Smart Tiles. *IEEE Sens. J.* **2017**, *17*, 469–479, doi:10.1109/jsen.2016.2625099.

20. Li, Y.; Ho, K.C.; Popescu, M. A microphone array system for automatic fall detection. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 1291–1301, doi:10.1109/TBME.2012.2186449.
21. Su, B.Y.; Ho, K.C.; Rantz, M.J.; Skubic, M. Doppler radar fall activity detection using the wavelet transform. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 865–875, doi:10.1109/TBME.2014.2367038.
22. Shiba, K.; Kaburagi, T.; Kurihara, Y. Fall Detection Utilizing Frequency Distribution Trajectory by Microwave Doppler Sensor. *IEEE Sens. J.* **2017**, *17*, 7561–7568, doi:10.1109/jsen.2017.2760911.
23. Wang, Y.; Wu, K.; Ni, L.M. WiFall: Device-Free Fall Detection by Wireless Networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 581–594, doi:10.1109/tmc.2016.2557792.
24. Wang, H.; Zhang, D.; Wang, Y.; Ma, J.; Wang, Y.; Li, S. RT-Fall: A Real-Time and Contactless Fall Detection System with Commodity WiFi Devices. *IEEE Trans. Mob. Comput.* **2017**, *16*, 511–526, doi:10.1109/tmc.2016.2557795.
25. Kido, S.; Miyasaka, T.; Tanaka, T.; Shimizu, T.; Saga, T. Fall detection in toilet rooms using thermal imaging sensors. In Proceedings of the 2009 IEEE/SICE International Symposium on System Integration (SII), Tokyo, Japan, 29 January 2009; pp. 83–88.
26. Yu, M.; Rhuma, A.; Naqvi, S.M.; Wang, L.; Chambers, J. A posture recognition based fall detection system for monitoring an elderly person in a smart home environment. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 1274–1286, doi:10.1109/TITB.2012.2214786.
27. Mirmahboub, B.; Samavi, S.; Karimi, N.; Shirani, S. Automatic monocular system for human fall detection based on variations in silhouette area. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 427–436, doi:10.1109/TBME.2012.2228262.
28. Agrawal, S.C.; Tripathi, R.K.; Jalal, A.S. Human-fall detection from an indoor video surveillance. In Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; pp. 1–5.
29. Poonsri, A.; Chiracharit, W. Improvement of fall detection using consecutive-frame voting. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; pp. 1–4.
30. 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, doi:10.1109/JBHI.2014.2304357.
31. Bian, Z.P.; Hou, J.; Chau, L.P.; Magnenat-Thalmann, N. Fall detection based on body part tracking using a depth camera. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 430–439, doi:10.1109/JBHI.2014.2319372.
32. Angal, Y.; Jagtap, A. Fall detection system for older adults. In Proceedings of the 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), Pune, India, 2–3 December 2016; pp. 262–266.
33. Ozcan, K.; Velipasalar, S.; Varshney, P.K. Autonomous Fall Detection with Wearable Cameras by Using Relative Entropy Distance Measure. *IEEE Trans. Hum.-Mach. Syst.* **2016**, doi:10.1109/thms.2016.2620904.
34. Ratner, P. *3-D Human Modeling and Animation*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2003; p. 336.
35. Free Video Tutorial: Anatomy: Scale & Proportion. Available online: <http://mimidolls.com/Video/Anatomy/Anatomy.php> (accessed on 29 September 2018).
36. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
37. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000**, *25*, 120, 122–125.
38. Chair Fail Gif by Cheezburger. Available online: <https://giphy.com/gifs/cheezburger-fail-fall-hU9THBubzgERW> (accessed on 5 September 2018).
39. Falls in Elderly People 5/5. Available online: https://www.youtube.com/watch?v=Od_RgAP8ojk (accessed on 5 September 2018).
40. Falling Out of Chairs! Available online: <https://www.youtube.com/watch?v=O-Ys7Q0rf34> (accessed on 5 September 2018).
41. CCTV Shows Drunk Girl Fall over on Path, Face First into the Soil. Available online: https://www.youtube.com/watch?v=cDZHS0W_LjY (accessed on 5 September 2018).
42. Funny People Falling on Ice Compilation. Available online: <https://www.youtube.com/watch?v=VgAWls11pco> (accessed on 5 September 2018).
43. Falls in Elderly People 1/5. Available online: <https://www.youtube.com/watch?v=p5i4z3sNaKM> (accessed on 5 September 2018).

44. Falls in Elderly People 2/5. Available online: https://www.youtube.com/watch?v=1IsM08Sh_wg (accessed on 5 September 2018).
45. Falls in Elderly People 3/5. Available online: <https://www.youtube.com/watch?v=3mDmkOxprN0> (accessed on 5 September 2018).
46. Falls in Elderly People 4/5. Available online: <https://www.youtube.com/watch?v=0VqvZGhK1o8> (accessed on 5 September 2018).
47. Caught on CCTV—The Big Slip. Available online: <https://www.youtube.com/watch?v=12bbjv8pEvA> (accessed on 5 September 2018).
48. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S., et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3296–3297.
49. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
50. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
51. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149, doi:10.1109/TPAMI.2016.2577031.
52. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
53. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
55. Zoph, B.; Le, Q.V. Neural Architecture Search with Reinforcement Learning. *arXiv* **2016**, arXiv:1611.01578.



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).