future internet

Article

Enhanced Fall Detection Using YOLOv7-W6-Pose for Real-Time Elderly Monitoring

Eugenia Tîrziu [1,2], Ana-Mihaela Vasilevschi [1], Adriana Alexandru [1,3,*] and Eleonora Tudora [1,*]

1 National Institute for Research and Development in Informatics, 011455 Bucharest, Romania; eugenia.tirziu@ici.ro (E.T.); ana.vasilevschi@ici.ro (A.-M.V.)
2 Doctoral School of Computer Science, Information Technology and Systems Engineering, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania
3 Faculty of Electrical Engineering, Electronics and Information Technology, Valahia University of Targoviste, 130004 Targoviste, Romania
* Correspondence: adriana.alexandru@ici.ro (A.A.); eleonora.tudora@ici.ro (E.T.)

Abstract: This study aims to enhance elderly fall detection systems by using the YOLO (You Only Look Once) object detection algorithm with pose estimation, improving both accuracy and efficiency. Utilizing YOLOv7-W6-Pose's robust real-time object detection and pose estimation capabilities, the proposed system can effectively identify falls in video feeds by using a webcam and process them in real-time on a high-performance computer equipped with a GPU to accelerate object detection and pose estimation algorithms. YOLO's single-stage detection mechanism enables quick processing and analysis of video frames, while pose estimation refines this process by analyzing body positions and movements to accurately distinguish falls from other activities. Initial validation was conducted using several free videos sourced online, depicting various types of falls. To ensure real-time applicability, additional tests were conducted with videos recorded live using a webcam, simulating dynamic and unpredictable conditions. The experimental results demonstrate significant advancements in detection accuracy and robustness compared to traditional methods. Furthermore, the approach ensures data privacy by processing only skeletal points derived from pose estimation, with no personal data stored. This approach, integrated into the NeuroPredict platform developed by our team, advances fall detection technology, supporting better care and safety for older adults.

Keywords: vision-based fall detection; YOLOv7; pose estimation; elderly; real-time alert; privacy-preserving

check for updates

Citation: Tîrziu, E.; Vasilevschi, A.-M.; Alexandru, A.; Tudora, E. Enhanced Fall Detection Using YOLOv7-W6-Pose for Real-Time Elderly Monitoring. *Future Internet* **2024**, *16*, 472. https://doi.org/10.3390/fi16120472

Academic Editor: Petros Patias

Received: 20 November 2024
Revised: 16 December 2024
Accepted: 17 December 2024
Published: 19 December 2024

1. Introduction

Global population aging is a demographic phenomenon marked by the increase in the proportion of elderly in the total population. According to the United Nations, the global population of people aged 65 and over is expected to double by 2050, reaching 1.5 billion [1].

As the population ages, the incidence of age-related health problems is increasing. In addition, aging brings with it increased frailty and a greater risk of domestic accidents, among which falls are particularly significant. According to data from the World Health Organization (WHO), there are over 684,000 catastrophic falls globally each year, with most victims being older than 60 [2]. In a world of ever-evolving technology, elderly care has become an area of major interest. One of the most important aspects of this area is the detection of falls, where falls represent a significant risk to the health and well-being of older people [3].

The aging of the population puts pressure on healthcare systems by raising the need for services, making labor shortages worse, and increasing the risk of both acute and chronic illnesses, including injuries from falls.

The integration of advanced technologies in elderly care is revolutionizing the industry, bringing significant improvements in their safety, efficiency, and quality of life [4]. By leveraging artificial intelligence and prioritizing user-centered design principles, we can develop more effective and user-friendly solutions [5]. These advancements not only address technical shortcomings but also contribute to a more supportive healthcare ecosystem for aging populations worldwide [6].

Fall detection systems are essential for the elderly, particularly those living independently or with minimal supervision [7]. These systems are designed to detect falls promptly and alert caregivers or emergency services, reducing the time to medical intervention and potentially preventing serious injuries [8]. However, despite advancements in technology, existing fall detection systems still face challenges and limitations that hinder their effectiveness, such as (1) a high rate of false alarms [9], which can lead to user frustration and loss of trust in the system; (2) high costs [10], limiting their accessibility for elderly individuals with low incomes; and (3) real-time detection limitations, delaying necessary interventions [11].

Fall detection technologies can be divided into two main categories: computer vision-based approaches and non-computer vision-based approaches. Non-computer vision-based methods such as accelerometers and vibration sensors detect falls using sound, vibrations, and human body movements.

Many people find accelerometers inconvenient to wear, and these sensors are susceptible to background noise, leading to false alarms. To overcome these limitations, computer vision-based fall detection methods are gaining popularity, as they use information from images and videos [7]. When developing a fall detection system for the elderly, the choice of object detection architecture is essential. The system must be able to quickly and accurately detect falls to allow immediate intervention.

A brief comparison between wearable technologies and vision-based systems is presented in Table 1.

Table 1. Comparison between wearable technologies and vision-based systems for fall detection of elderly.

Aspect	Vision-Based Systems	Wearable Technologies
Accuracy	Detects falls in the context of the surroundings; false positives, illumination, and occlusion problems.	Direct body data (posture, motion) could contain errors or false positives depending on device placement.
Privacy	Less intrusive; only records health and mobility information.	Less intrusive; only records health and mobility information.
Cost	Expensive initial setup; more economical in shared environments (e.g., care facilities).	Individual users can afford it, but long-term maintenance raises expenses.
Environmental Awareness	Provides environmental context, detecting falls and hazards like obstacles or slippery floors.	Lacks awareness of surroundings (e.g., slippery floors or obstacles).
Scalability	Costly for individual homes, but more scalable in communal or institutional age care settings.	Scalable for individual use; higher costs for large populations.

Three of the most popular architectures used for object detection are YOLO (You Only Look Once) [12], SSD (Single-Shot MultiBox Detector) [13], and RetinaNet [14]. Each of these architectures offers distinct advantages and disadvantages in terms of speed, accuracy, and complexity of implementation. YOLO is known for its speed. It processes images in a single forward pass through the network (having the capability to perform people detection and pose estimation simultaneously), making it extremely fast and suitable for real-time applications [15]. SSD strikes a balance between speed and accuracy. It uses a convolutional neural network to predict bounding boxes and classes directly from feature maps at multiple scales. RetinaNet is known for its accuracy due to the Focal Loss function, which improves detection performance for hard-to-detect objects, such as small objects or imbalanced datasets [16]. To better understand the differences between these architectures and determine the most appropriate choice for real-time drop detection, we present a comparison table below (Table 2). The sets of values for YOLO, SSD, and RetinaNet are presented based on references [12–16] and are obtained by using various types of hardware.

Table 2. Comparison of three of the most popular architectures used for object detection.

Feature	YOLO	SSD	RetinaNet
Used hardware	NVIDIA GeForce GTX 1080Ti GPU, 32 GB of RAM, Intel Core i7 processor		
Speed (frames per second FPS)	~40–84 FPS	~20–46 FPS	~5–17 FPS
Accuracy	Good, struggles with small objects	Good, better for various sizes	High, especially for small objects
Complexity	Simple to implement	Moderate complexity	More complex
Use Case	Real-time applications	Real-time applications	Best for accuracy, not real-time
Special Features	Unified model, fast processing	Multi-scale detection	Focal loss for class imbalance

YOLO stands out as the most suitable architecture for a fall detection system due to its exceptional speed, which enables near-immediate detection and reaction to critical events. The simplicity of implementation, the ability to perform well in low-light conditions, and the low resource requirements make YOLO an optimal solution for real-time monitoring of the elderly. While SSD and RetinaNet offer advantages in other areas, YOLO provides the best balance for a fast and efficient fall detection system [17].

The integration of the YOLO algorithm into elderly care systems offers numerous benefits, significantly enhancing the quality of care provided to seniors [18]. One of its primary advantages is its real-time monitoring capability.

YOLO, a cutting-edge object detection model, redefines object detection as a single regression problem by mapping image pixels directly to bounding box coordinates and class probabilities [19]. Unlike traditional methods that apply classifiers across multiple image locations and scales, YOLO uses a single neural network for the entire image. It divides the image into regions and predicts bounding boxes and probabilities for each, allowing for extremely fast processing. This capability makes real-time applications, such as fall detection, not only feasible but also highly effective in dynamic environments [20–25]. The model's precision minimizes false positives and negatives, ensuring caregivers are alerted only for genuine emergencies, thereby improving response times and care quality.

One of the most important benefits of the YOLO algorithm is its *capability of real-time object detection*, which is essential for elderly care, where the immediate detection of falls can trigger prompt responses, potentially preventing severe injuries and complications. The speed of YOLO (see Table 2) ensures that caregivers are alerted without delay, enabling swift intervention.

Beyond speed, YOLO is renowned for its *accuracy*, minimizing false positives and negatives in object detection. This precision is important in fall detection systems, ensuring caregivers are alerted only for genuine emergencies, reducing false alarms, and improving care quality [25].

Another significant advantage of YOLO is *its versatility in handling multiple objects within a single frame*. Elderly care environments are often dynamic, involving multiple individuals and objects. YOLO's ability to differentiate between objects and people ensures focused monitoring, even in complex and cluttered settings. This robustness makes YOLO-based systems reliable and adaptable to diverse environments.

YOLOv7, a more advanced version, incorporates architectural improvements like efficient convolutional neural networks (CNNs), multi-scale detection for varying object sizes, and robust training techniques such as data augmentation. These upgrades enhance its speed, accuracy, and ability to generalize across diverse scenarios [26–30]. When combined with pose estimation, YOLOv7 enables a detailed analysis of human posture and movement, which is critical for distinguishing between routine actions and falls, even in challenging conditions such as poor lighting or unusual camera angles [31–33].

CUDA Python further optimizes YOLOv7 by leveraging Nvidia GPUs, significantly accelerating the computational demands of deep learning, particularly for real-time applications [34,35]. This is essential for tasks requiring high-speed and reliable processing, such as fall detection in elderly care, where immediate responses are paramount.

The hybrid YOLOv7-W6-Pose model enhances these capabilities by integrating advanced object detection with detailed pose estimation. YOLOv7-W6, a wider version, balances speed and accuracy, making it ideal for real-time scenarios. When paired with pose estimation, it identifies key points like joints and limbs, enabling nuanced interpretations of body movements and postures [36,37]. This combination not only detects falls but also captures the context and dynamics leading to them, offering a comprehensive monitoring solution.

The implementation of YOLOv7-W6-Pose involves several steps:

1. *Image Preprocessing:* Input images are resized while maintaining aspect ratios through a letterbox function to avoid distortions.
2. *Forward Pass:* Images are processed through the network, which predicts bounding boxes, class probabilities, and key point locations for each grid cell.
3. *Output Generation:* The model outputs bounding boxes encapsulating detected objects and maps of key body points such as the head, neck, shoulders, elbows, hips, and ankles.
4. *Redundant Detection Elimination:* Non-Maximum Suppression (NMS) eliminates redundant detections, retaining the most confident predictions for improved accuracy.
5. *Key point Analysis:* The generated skeleton map allows for detailed posture and movement analysis, enabling accurate fall detection and differentiation from normal activities.

In aged care settings, where continuous monitoring is often required, an efficient fall detection system can reduce the risk of unattended falls. The proposed system offers reliable real-time detection, supporting caregivers and healthcare professionals in delivering timely interventions in response to incidents [38].

The objective of this study was to develop a real-time fall detection system by using the YOLO object detection algorithm with pose estimation. This system was developed under the ongoing project entitled "Advanced Artificial Intelligence Techniques in Science and Applications", which focuses on advancing mental health monitoring using the NeuroPredict platform. Pre-trained models were implemented to improve the accuracy of fall detection and to expedite the development process, enhancing overall system performance.

The main contributions of this study are as follows:

- Development of a vision-based fall detection approach: The proposed system enhances the detection accuracy and reliability of fall detection by utilizing advanced computer vision techniques.
- By integrating a pre-trained YOLOv7-W6-Pose, the system accurately estimates human key points, enabling precise monitoring of posture and movement.
- Creation of a comprehensive fall dataset: This dataset includes a wide variety of scenarios, compiled from videos sourced both from the internet and through original

recordings. This diverse dataset enables the robust training and evaluation of the proposed model.

This paper is organized into the following sections: the Introduction presents the primary objectives and the scope of the study, highlighting the integration of human detection and pose estimation techniques to improve fall detection accuracy and efficiency. Related work contains a brief presentation of relevant emerging approaches. The Materials and Methods describes the methodologies and tools utilized in this study, including materials, data collection and preprocessing, the Fall Detection Algorithm for pose estimation, and the Alert System developed for achieving the paper's objectives. The Results section presents the analysis of the performance of the system and the benefits of using a real-time fall detection system, particularly in the context of elderly care. The Discussion and Conclusions contain a brief description of the findings and their implications, limitations of the work, and future research directions.

2. Related Work

A review of the relevant literature on fall detection and related technologies reveals a rapidly evolving landscape marked by advancements in sensor technology, machine learning algorithms, and telecommunication systems [39]. Researchers and practitioners alike are increasingly recognizing the importance of early fall detection in mitigating the adverse consequences of falls among the elderly population. Recent approaches from the latest studies in this field are illustrated in Figure 1.

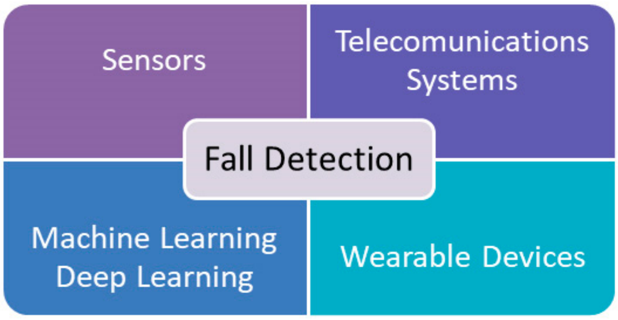

Figure 1. Recent approaches to fall detection.

Sensor Technologies for Fall Detection: Accelerometers and gyroscopes, which are inertial sensors, are commonly used for fall detection due to their ability to measure motion and orientation changes, with research indicating their effectiveness in accurately detecting falls while minimizing false alarms [40,41]. Additionally, pressure sensors, including floor-based pressure sensors and wearable pressure-sensitive mats, have shown promise in detecting falls by identifying abrupt changes in pressure distribution upon impact with the ground [42]. Furthermore, depth-sensing cameras, such as Microsoft Kinect, have been explored for fall detection by analyzing skeletal movements and changes in body posture, offering the potential for accurate fall detection in home environments [9,43].

Machine Learning and Deep Learning Algorithms: Support Vector Machine (SVM) classifiers have been widely used for fall detection, demonstrating high accuracy in distinguishing between fall and non-fall events based on sensor data features [44]. Additionally, Artificial Neural Network (ANN) models have shown promising results in fall detection by learning complex patterns from sensor data, with deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) also being applied successfully. Furthermore, ensemble learning methods, such as random forests and gradient boosting, have emerged as effective techniques for improving fall detection performance by combining multiple classifiers [45,46].

Telecommunication Systems: Smartphones, with their built-in sensors and communication capabilities, offer a convenient platform for fall detection applications. Researchers have explored various smartphone-based solutions, including apps and wearable accessories, for continuous monitoring and emergency alerting [47].

Wearable Devices: Similarly, smartwatches and wearable devices equipped with accelerometers and other sensors have gained traction for fall detection among older adults, offering the advantage of continuous monitoring and seamless integration into daily routines [48].

Challenges and Future Directions: As fall detection technologies become more pervasive, ensuring user privacy and addressing ethical concerns surrounding data collection and sharing remain paramount. Engaging end-users, including older adults and caregivers, in the design and evaluation of fall detection systems is essential to ensure usability, acceptance, and long-term adoption [42]. Furthermore, additional research is required to validate the efficacy of fall detection technologies in real-world settings and diverse populations, considering factors such as environmental variability and user mobility [46].

In previous approaches to fall detection, several key issues and challenges have been identified, which have prompted ongoing research and innovation in the field. Here, we summarize some of the main issues along with references to relevant studies:

- *False Alarms and Sensitivity*: Many fall detection systems suffer from high rates of false alarms, leading to user frustration and reduced system reliability [11].
- *Environmental Variability*: Fall detection systems may struggle to perform reliably in diverse environmental conditions, such as different lighting, flooring surfaces, or room layouts [49].
- *User Acceptance and Wearability*: Wearable fall detection devices may face challenges related to user acceptance, comfort, and compliance, particularly among older adults [50].
- *Privacy and Ethical Considerations*: Fall detection systems may raise concerns related to user privacy, data security, and the ethical use of personal health information [51].
- *Real-World Validation and Deployment*: Despite promising results in controlled settings, many fall detection systems lack validation in real-world environments and may face challenges in practical deployment.

Addressing these challenges is essential for the development of reliable and effective fall detection systems for older adults. Ongoing research efforts focus on overcoming these obstacles through advances in sensor technology, machine learning and deep learning algorithms, and user-centered design approaches [7].

3. Materials and Methods

The objective of this study was to develop and evaluate a real-time fall detection system using the YOLO object detection algorithm for pose estimation.

3.1. Materials

The primary hardware was the INSTA360 GO3, 64GB, Wi-Fi camera, chosen for its affordability, ease of use, and advanced safety monitoring features. It records in up to 2.7K resolution, providing clear, stable images even during motion, and syncs easily with an app for real-time video streaming and analysis.

The system was implemented in Python, utilizing several key libraries:

- OpenCV for video capture and image processing [52].
- PyTorch for machine learning, enabling real-time applications [53].
- Torchvision for datasets and image transformations [54].
- NumPy for handling multi-dimensional arrays and mathematical operations [55].

For our research, we used a GPU with the following characteristics: nVidia GeForce RTX 4070, 12 GB, GDDR6. This parallelism allows the YOLOv7 model to process video frames quickly and accurately, ensuring immediate responses to potential falls. The integration of CUDA Python with the system enabled us to achieve the necessary performance levels to meet the real-time requirements of our application.

For detecting and estimating human body key points such as shoulders, elbows, knees, and feet, we utilized YOLOv7-W6-Pose. YOLOv7-W6-Pose is specifically designed to integrate these two tasks, allowing for efficient analysis of human posture in real-time applications.

The model setup and training process were performed according to the documentation available in the official YOLOv7 repository. The model's ability to accurately predict these key points is important for fall detection applications.

We sourced pre-trained models directly from the official YOLOv7 repository on GitHub [56], which provides access to various versions optimized for different scenarios. YOLOv7-W6-Pose allows us to achieve simultaneous person detection and key point estimation, which is essential for our fall detection application.

3.2. Study Design

The system was designed to capture video streams from a camera and process these images to detect human poses and identify potential falls. Upon detecting a fall, it sends an alert via a Telegram bot [57] to promptly inform caregivers or family members when a fall is detected, ensuring timely intervention.

The system begins by capturing video frames from a monitoring setup, as illustrated in Figure 2. These frames are then preprocessed to enhance their quality and suitability for analysis. The next step involves checking for the presence of a person in the preprocessed frames. If no person is detected, the system captures the next video frame. When a person is detected, the system proceeds to the pose estimation phase, applying algorithms to identify key body points and estimate their posture. Here, the system applies pose estimation and runs a fall detection algorithm on the video recordings. The system then evaluates whether a fall has been detected. If no fall is detected, it loops back to continue capturing and processing video frames. If a fall is detected, the workflow transitions to the alert mechanism.

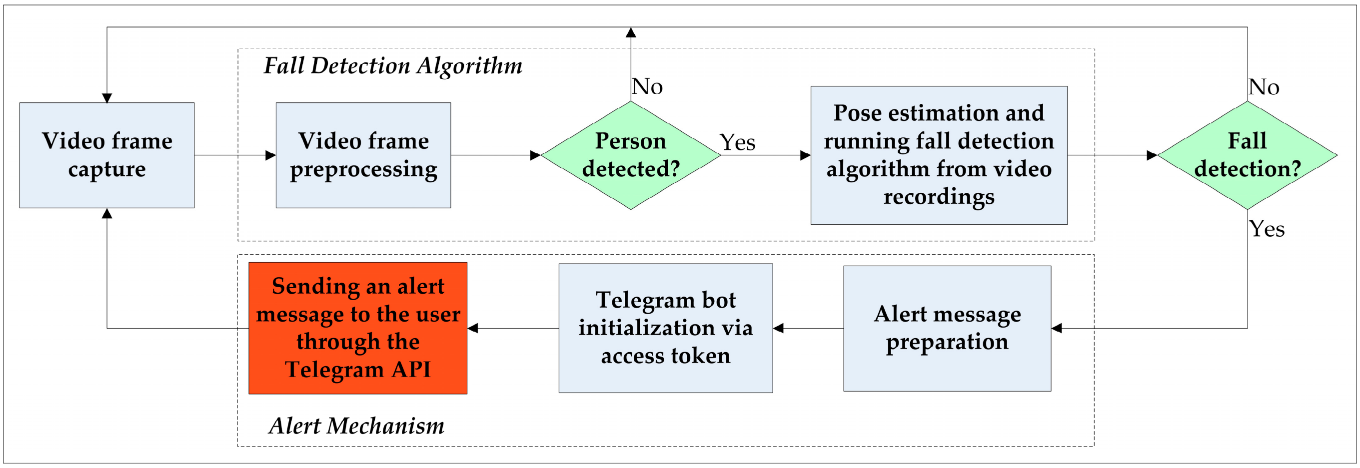

Figure 2. Flowchart of the fall detection system.

In the alert mechanism, a Telegram bot is initialized using an access token and prepared to send notifications. Once a fall is detected, an alert message is composed, detailing the event. The message is then sent to the user via the Telegram API, notifying them of the incident.

To ensure users' privacy is preserved, it is important to note that the information processed by the system is not saved. Video frames are analyzed exclusively in real time, and none of the captured frames or data are stored after processing. This approach prevents unauthorized storage and access to sensitive data while respecting data privacy and security standards.

Because YOLOv7-W6-Pose recognizes skeletal key points rather than gathering or storing comprehensive visual data about individuals, it allows posture estimation from webcam videos in a privacy-conscious way. Since there is insufficient information in the key points to allow for visual identification of an individual, this method effectively anonymizes information. Pose estimation promotes privacy by limiting the acquisition of individually identifying information by concentrating on posture and body movement rather than looks.

To preserve privacy, anonymized key point data can be safely shared with monitoring systems or caregivers. YOLOv7-W6-Pose is ideal for settings like smart homes or elder care facilities where the goal is to identify significant occurrences (like falls) while maintaining privacy standards because it allows for selective monitoring without continuous video recording.

Figure 2 underscores the system's capability to provide continuous and reliable monitoring, thereby contributing to improved safety and response times in elderly care scenarios.

To better understand how this system operates, we will analyze the specific algorithm that underpins its functionality. Our algorithm analyzes a sequence of frames (not just a single static frame). Fall detection relies on tracking the movement of the human skeleton over time by analyzing, for a sequence of human pose skeletons, the position and orientation of key points between consecutive frames to detect abrupt changes that could indicate a fall. This dynamic approach ensures higher accuracy, as falls typically involve a sudden transition from a vertical to a near-horizontal position. This approach provides higher accuracy because a fall usually involves a rapid and sudden transition of the body from a vertical to a near-horizontal position. The detailed workings of this advanced fall detection system are presented in Supplementary Material S1.

For this system, the output can be single/multiple and is represented by the number of falls for which an alert is sent.

3.3. Data Collection and Preprocessing

Video Capture: Real-time video data were captured using a web camera, accessed through the OpenCV library. OpenCV provided tools to read frames from the video feed sequentially, ensuring a continuous stream of image data for real-time processing. The use of OpenCV enabled the efficient extraction of each frame from the video stream, which is essential for real-time monitoring and fast drop detection.

Image Preprocessing: Each captured frame was resized to 960 $\times$ 960 pixels using a letterbox function to preserve the aspect ratio and avoid distortion. The frames were then converted into tensors and normalized using the ToTensor function from the torchvision library. This step was essential to prepare the frames for input into the PyTorch model, which operates on tensors.

Through this preprocessing, each video was correctly formatted for the YOLOv7-W6-Pose estimation model, enabling efficient real-time fall detection. Thus, the system can quickly and efficiently analyze images to detect possible falls in real time.

3.4. Pose Estimation and Fall Detection Algorithm

3.4.1. Pose Estimation with YOLOv7-W6-Pose

The model processes each frame of the video to locate these points accurately, enabling the system to monitor the movements and postures of individuals continuously.

To detect falls, an algorithm was developed that analyzes the positions of the identified key points. The algorithm works as follows:

- *Position Analysis*: The algorithm tracks the relative positions of key body points, focusing particularly on the shoulders, torso, and feet. By continuously monitoring these positions, the algorithm can detect changes that may indicate a fall.
- *Distance Calculations*: It calculates the distances between these key points to assess the body's posture. For example, the algorithm measures the distance between the shoulders and the feet.
- *Fall Detection Criteria*: Significant changes in these measurements can suggest a fall. For instance, a noticeable drop in the positions of the shoulders relative to the feet indicates that the person has fallen.

- *Thresholds and Conditions*: To accurately differentiate between normal activities and falls, the algorithm employs thresholds and conditions that have been fine-tuned using empirical data. This fine-tuning involved analyzing a dataset of various activities and fall events to determine the optimal threshold values for distances. These thresholds ensure that the algorithm can robustly identify falls without generating false alarms.

Using a fixed threshold might not be effective because people have different body sizes. Setting a dynamic threshold based on the person's size and proportions provides a personalized threshold for each pose, enhancing the accuracy of fall detection. Our system integrates YOLOv7-W6-Pose and can automatically adjust thresholds by dynamically calculating body proportions, movement patterns, and key point positions based on the detected skeletons.

- *Fine-Tuning and Optimization*: The algorithm was iteratively improved by testing it with real-world data and adjusting the thresholds and conditions based on the observed performance. This process ensured that the algorithm could reliably detect falls while minimizing false positives. By leveraging empirical data, the algorithm was calibrated to distinguish between typical daily activities and actual fall events accurately.

The pose estimation provided by the YOLOv7-W6-Pose model, combined with the custom fall detection algorithm, forms a robust system for monitoring and detecting falls. The careful analysis of key point positions and the fine-tuning of detection criteria ensure high accuracy in identifying falls.

3.4.2. Fall Detection Algorithm

The fall detection algorithm uses a geometric approach to analyze the positions of different parts of a person's body and determine whether they have fallen. Specifically, it relies on the relationship between the positions of the shoulders, torso, and feet, correlating these positions to assess whether the current posture indicates a fall. To better understand this process, it is essential to detail how these coordinates are established and compared, as well as the thresholds that define a fall.

1. The algorithm begins by *extracting the coordinates of key points on the body*:
   - Left and right shoulders: $S_l(x_l,y_l)$, $S_r(x_r,y_r)$.
   - Torso (both left and right sides): $T_l(x_{Tl},y_{Tl})$, $T_r(x_{Tr},y_{Tr})$.
   - Feet (left and right): $F_l(x_{Fl},y_{Fl})$, $F_r(x_{Fr},y_{Fr})$.
2. After obtaining these coordinates, an intermediate value is first calculated, called the "*length factor*". This factor is computed based on the geometric distance between the position of the left shoulder and the left torso, expressed using the Euclidean distance formula. This factor is later used to determine a relationship between the various parts of the body and evaluate their relative position.

$$L_{factor} = \sqrt{\left(x_l - x_{Tl}\right)^2 + \left(y_l - y_{Tl}\right)^2}, \tag{1}$$

This factor adjusts the detection thresholds so that the algorithm is applicable regardless of people's height or proportions.

3. *Thresholds* used for fall detection:
   - Height of the shoulders relative to the feet: This checks whether the vertical coordinate of the left (or right) shoulder is lower than the vertical coordinate of the left (or right) foot, adjusted by the length factor. In a normal posture, the shoulders should be above the feet, but during a fall, they may end up lower or at a similar level to the feet. The algorithm considers a possible fall if the shoulders drop to a level close to the feet.
     The threshold can be defined as follows:

$$y_l \leq y_{Fl} + \alpha \cdot L_{factor}, \tag{2}$$

where α is a small adjustment factor to account for natural variations of the body in a normal position.

– Difference between the body's width and height: The algorithm calculates the difference between the vertical (height) and horizontal (width) dimensions of the bounding box that encloses the body. If the height becomes smaller than the width (i.e., the body is "stretched" horizontally rather than vertically), it is an indication that the person has fallen.

The calculation of body dimensions can be performed as follows:
Body height:

$$H_{body} = |y_l - y_{Fl}|, \tag{3}$$

Body width (the distance between the shoulders or between the sides of the chest):

$$W_{body} = |x_l - x_r|, \tag{4}$$

The threshold for detecting a fall could be:

$$H_{body} < W_{body}, \tag{5}$$

4. *Fall decision*

If all conditions are met—meaning the shoulders and torso are at an abnormally low height compared to the feet, and the body has a greater horizontal than vertical dimension—the algorithm concludes that a fall has occurred. In this case, a positive fall signal is returned, along with the coordinates of the bounding box enclosing the body.

Key thresholds in the algorithm:

– Height of the shoulders and torso relative to the feet: This threshold is adjusted using the length factor to account for individual body proportions. Essentially, it checks if the shoulder and torso drop below an adjusted level relative to the feet.
– Difference between body dimensions: If the difference between the height and width of the body becomes negative (height is smaller than width), this signals that the person is lying down.

5. *Differentiation between a fall and a person who is simply lying down*.

To improve fall detection accuracy, we have differentiated between a fall and someone who is just lying down. One significant improvement is tracking the speed of movement. When a fall occurs, there is a rapid change in body position, so we calculated vertical speed using the movement of the shoulders, torso, or feet. If the vertical speed exceeds a specific threshold, it indicates a fall, as opposed to someone slowly lying down. This way, the algorithm can detect sudden drops, which are characteristic of falls, rather than more controlled movements like lying down. The speed of key body points is calculated by measuring the displacement between their positions in consecutive frames.

Another way to distinguish between falls and intentional lying down is to measure the angles between important body parts. When someone falls, their torso and legs frequently become more parallel to the ground, resulting in acute angles. In contrast, lying down involves a smoother, more controlled transition in body posture. A threshold of 45 degrees is used in the code. If the angle between the torso and legs drops below this value, it indicates that the person's body is approaching a horizontal position, which is typical of a fall [58]. For lying down, the transition is slower, and the body takes time to align in this way.

The alert system is a component designed to notify caregivers or family members immediately when a fall is detected. This mechanism ensures prompt intervention, which can be vital in preventing further injury or complications for the individual who has fallen. Our system includes an alert mechanism using the Telegram Bot API. When a fall is detected, the system sends an alert message to a predefined chat ID (Figure 3).

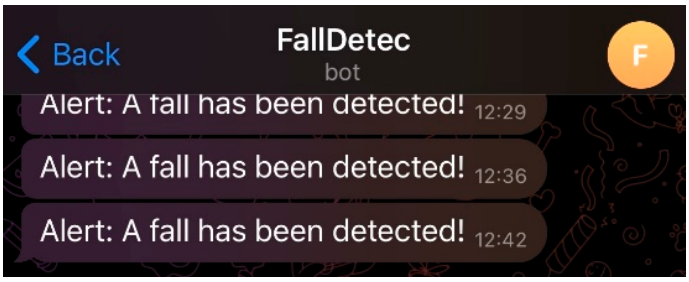

Figure 3. Real-time fall detection alerts.

3.4.3. Telegram Bot Integration

The Telegram Bot API allows the system to send automated messages to a predefined chat ID whenever a fall event is detected. This integration involves the following steps:

- *Bot Setup*: A Telegram bot is created and configured with a unique API token. This token is used to authenticate and send messages on behalf of the bot.
- *Chat ID Configuration*: The predefined chat ID, which corresponds to the recipient (caregiver or family member), is configured within the system. This ensures that alerts are directed to the correct individual or group.

3.4.4. Preventing Spam

To prevent spamming and ensure that alerts are meaningful, the system includes a mechanism to limit the frequency of messages. This is achieved by implementing the following logic:

- *Timestamp Recording*: Each time an alert is sent, the system records the current timestamp.
- *Minimum Interval Enforcement*: Before sending a new alert, the system checks the timestamp of the last sent message. If the current time is less than two minutes from the last alert, the system suppresses the new alert. This ensures that alerts are sent no more than once every two minutes, preventing repetitive notifications for the same event or minor, non-critical movements.

3.4.5. Asynchronous Messaging

The asynchronous nature of Telegram messaging is important for maintaining the system's real-time performance. Here is how it works:

- *Non-blocking Operations*: Sending messages via the Telegram Bot API is handled asynchronously, meaning the system does not wait for the message to be delivered before continuing with other tasks. This non-blocking operation allows the fall detection algorithm to continue processing video frames and detecting potential falls without interruption.
- *Real-Time Performance*: By ensuring that the alerting mechanism does not block the main processing loop, the system maintains its real-time performance. This is essential for continuous monitoring and the timely detection of falls.

3.5. Real-Time Video Processing

The system continuously processes video frames in real time. It captures frames from the camera, detects poses using the YOLOv7 model, evaluates fall conditions, sends alerts if necessary, and displays the results. This real-time processing loop ensures immediate responses to potential falls, enhancing the system's safety and monitoring capabilities. The loop runs until manually terminated, providing continuous surveillance and fall detection.

3.6. System Evaluation Procedure

As part of the development of an advanced fall identification system, we selected several free videos from online sources [59–61] for upload and analysis to validate the algorithm's effectiveness. These videos include daily activities such as walking, sitting, standing, and bending, as well as different types of falls, including forward, backward, and lateral falls, along with falls from varying heights and with different levels of impact. The results obtained after this preliminary stage were promising, demonstrating the system's ability to detect fall events with high accuracy. Subsequently, we downloaded the Le2i fall dataset [62] and tested the system's accuracy.

To extend validation and ensure that the system functions correctly in various scenarios and in real time, videos recorded with a webcam were additionally implemented and tested. These real-time tests allowed the evaluation of the system behavior under dynamic and unpredictable conditions, providing valuable opportunities to adjust and improve the algorithm.

The following types of tests were performed:

1. *Correct detection of persons*—to confirm its ability to accurately detect persons within the frame. This step is essential as it serves as the foundation for identifying subsequent actions, including falls. The real-time object detection capabilities of the YOLO algorithm were utilized to differentiate between persons and other objects in the environment (Figure 4a–c). In cases where no person is detected within the frame, the algorithm can confirm the absence of a person, thereby ensuring the accuracy of the monitoring process (Figure 4d).
2. *Correct identification of falling moments*—to accurately identify the moments when falls occur. This involved analyzing various fall scenarios, such as forward, backward, and sideways falls, and ensuring that the system could correctly classify these events. The goal was to minimize false positives (incorrectly identifying non-fall activities as falls) and false negatives (failing to detect actual falls) (Figure 4e). To evaluate both the false positive and false negative rates, we conducted tests that included a variety of activities of daily living that could be mistaken for falls. For instance, we included scenarios such as changing shoes, picking up objects from the floor, and performing exercises on the ground.

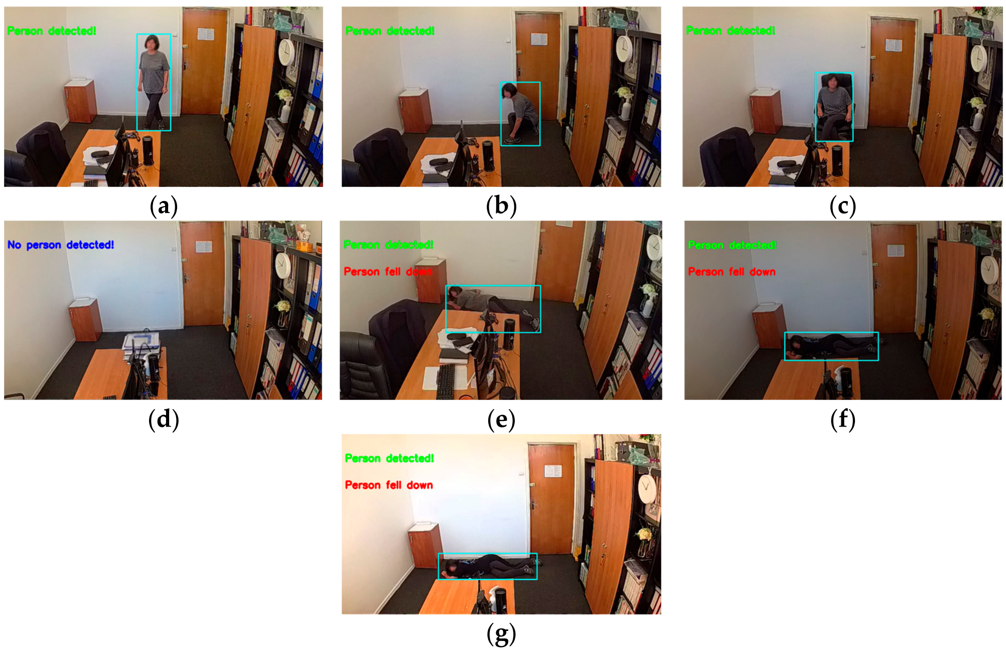

Figure 4. Captured frames from videos in diverse conditions: (**a**) identification of a person; (**b**) identification of a person bending over; (**c**) identification of a person sitting on the chair; (**d**) no person present; (**e**) identification of falling; (**f**) fall detection in environments with very low light levels; (**g**) fall detection in environments with intense lighting conditions.

3. *Evaluating model performance in a various lighting condition*—to assess how the model performs under different lighting scenarios, such as very poor or very bright lighting,

to check the limitations of the model. The algorithm was tested in extreme lighting conditions to assess its limitations:

- *Very Poor Lighting*: The system was tested in environments with very low lighting to evaluate its ability to detect falls accurately in situations where visibility is compromised. Despite the low light levels, the model managed to perform well in detecting falls, maintaining an acceptable level of accuracy. However, there were cases where the visibility was significantly reduced, leading to slight inaccuracies in pose estimation, especially when movements were subtle or fast. Still, the system showed a reasonable tolerance for dim lighting (Figure 4f).
- *Very Bright Lighting*: Similarly, tests in very bright lighting conditions, which can lead to overexposure or glare, were conducted to understand how the model handles such challenges. In these cases, the system showed good performance, with the ability to distinguish body movements even in intense lighting. However, glare or reflections on the surface occasionally created small disturbances, although the system was generally able to filter these out, maintaining a high detection accuracy. This suggests that while the system is robust in bright lighting, further refinement could be made to reduce the impact of intense light sources (Figure 4g).

4. Results

4.1. Analysis of Performance of the System in the Detection of Falls

The evaluation of the results can be better understood using a confusion matrix, which provides a comprehensive visualization of the performance of a classification algorithm. The confusion matrix is a fundamental tool in evaluating the performance of classification algorithms, providing a detailed breakdown of predictions versus actual outcomes across different classes. In the context of fall detection or any binary classification task, the confusion matrix helps quantify the model's accuracy, precision, recall (sensitivity), specificity, and other performance metrics.

To evaluate the performance of the fall detection system, extensive tests were performed using a confusion matrix that provides an overview of the algorithm's ability to correctly identify both falls and normal activities.

The confusion matrix for this scenario is represented in Figure 5:

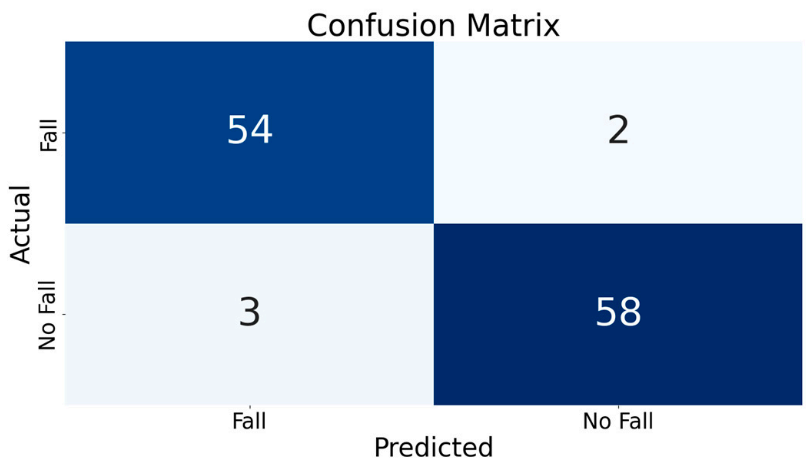

Figure 5. Confusion matrix.

Each component of the confusion matrix is explained below:

- *True Positives (TP)*: True positives are instances where the model correctly predicts the positive class. In fall detection, this would be when the system correctly identifies a fall event. In this case, there were 54 instances where falls were correctly identified by the model.

- *True Negatives (TN)*: True negatives are instances where the model correctly predicts the negative class. In fall detection, this would be when the system correctly identifies a non-fall activity. This represents 58 instances where the model correctly identified activities as non-falls.
- *False Positives (FP)*: False positives occur when the model predicts the positive class incorrectly. In fall detection, this would be when the system incorrectly identifies a fall when no fall has occurred. False positives are also known as Type I errors. In this scenario, there were three false alarms.
- *False Negatives (FN)*: False negatives occur when the model predicts the negative class incorrectly. In fall detection, this would be when the system fails to identify a fall when one has actually occurred. False negatives are also known as Type II errors. Here, two fall events were missed by the model.

Using these values, key metrics can be calculated to assess the performance of the algorithm, as illustrated in Table 3:

Table 3. Key metrics used for system evaluation.

Metrics	Formula	Value
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$ ($\times$100%)	95.7%
Precision (P)	$\frac{TP}{TP+FP}$ ($\times$100%)	94.7%
Recall (R)	$\frac{TP}{TP+FN}$ ($\times$100%)	96.4%
Specificity	$\frac{TN}{TN+FP}$ ($\times$100%)	95%
F1 score	$\frac{2\times P\times R}{P+R}$ ($\times$100%)	95.5

Interpretation and Application

- *High Recall*: Recall, also known as sensitivity or the true-positive rate, measures the proportion of actual positives that are correctly identified by the model. This exhibits a high recall (96.4%), indicating that it effectively detects a large majority of actual fall events.
- *Good Accuracy*: Accuracy measures the overall correctness of the model's predictions across all classes. The accuracy (95.7%) means that the model is correct in its predictions for many of the instances in the dataset, demonstrating its capability to make reliable assessments in the context of fall detection.
- *Good F1 Score*: The F1 score combines both precision and recall into a single metric, providing a balanced assessment of the model's performance in detecting falls. An F1 score of 95.5% indicates that the fall detection model achieves a robust balance between precision and recall, making it suitable for practical deployment in scenarios where accurate detection and minimal false alarms are important. A higher F1 score indicates better overall performance in terms of both precision and recall.
- *Good Precision*: Precision measures the proportion of true-positive predictions among all positive predictions made by the model. The precision (94.7%) indicates that when the model predicts a fall, it is correct most of the time.
- *Good Specificity*: Specificity measures the proportion of actual negatives that are correctly identified by the model. The specificity (95%) shows that the model correctly identifies non-fall events in a majority of cases.

These metrics, derived from the confusion matrix, provide a detailed understanding of the algorithm's performance in detecting falls and distinguishing them from non-falls.

To further contextualize these results, it is important to compare our system with other existing fall detection solutions. By analyzing the performance metrics—such as accuracy, precision, recall, specificity, and F1 score—of similar systems, the strengths and potential areas for improvement in our approach were highlighted (Table 4). We utilized the Le2i fall dataset for our evaluations, which comprises a total of 130 video sequences. The dataset consists of video sequences from four environments: Home, Coffee room, Office, and Lecture room. Each video is formatted at a resolution of 320 × 240 and 25 FPS. For each video, a corresponding ground truth file is provided. These files contain the frame numbers marking the start and end of a fall, along with the bounding box details: height, width, and the coordinates of the center for each frame. No augmentation methods were applied to the dataset. The dataset includes 99 sequences that depict falls and 31 that showcase various normal activities. The falls in the dataset are categorized into several types (forward falls, backward falls, lateral falls, falls from various heights, and falls with different impact levels). The normal activities included in the dataset are walking, sitting down or standing up from a chair, bending over to pick up objects, changing shoes, and performing stretches or exercises. The dataset presents challenges related to variable lighting conditions and occlusions.

Table 4. A comparison of our proposed method with other studies.

Model	Accuracy	Precision	Recall	Specificity	F1 Score
Chamle et al. [63]	79.30%	84.30%	73.07%	79.40%	78.28%
Alaoui et al. [64]	90.90%	94.56%	81.08%	90.84%	87.30%
Poonsri et al. [65]	86.21%	89.1%	93.18%	64.29%	91.1%
Our method	**96.15%**	**97%**	**97.98%**	90.32%	**97.48%**

Our proposed model outperforms other similar methods in terms of accuracy, specificity, precision, and F1 score, while also achieving above-average recall. These outcomes highlight the efficacy of our approach to detecting human falls and suggest its potential applicability in real-world scenarios. Our approach has been tested in laboratory conditions.

4.2. Analyzing the Performance of Using the YOLO Algorithm Based on Pose Estimation in the Context of Elderly Care

In the context of elderly care, the implementation of the YOLO algorithm provides a number of key advantages for detecting falls and improving the overall safety and quality of life:

- *Real-time monitoring*: The YOLO algorithm enabled the rapid and accurate detection of falls in real time, facilitating immediate intervention and reducing the risk of complications for the elderly. Specifically, the average pose estimation time was 361.4 milliseconds, the average fall detection time was 0.06162 milliseconds, and the average alarm transmission time was 0.0336 milliseconds.
- *Improved safety*: By proactively identifying hazards, the algorithm helped prevent accidents, thereby increasing the safety of residents.
- *Resource Optimization*: Reducing the need for constant human supervision has allowed caregivers to focus on more critical and personalized tasks, improving the quality of care.
- *Respect for privacy*: The algorithm provided selective monitoring, ensuring the privacy and dignity of the elderly.
- *Adaptability and customization*: The YOLO algorithm's flexibility and pose estimation allowed the system to be customized to recognize specific scenarios and objects, thus improving its effectiveness and applicability in various elderly care contexts.

The obtained results demonstrate that the use of the YOLO algorithm together with pose estimation brings significant improvements in the detection of falls and the quality of care for the elderly. The system has proven to be robust and effective under various conditions, providing a reliable solution for monitoring and ensuring the safety of the elderly. The continued development and adaptation of these technologies will play an important role in improving the lives and health of older people, providing them with a greater level of independence.

5. Discussion and Conclusions

The integration of the YOLO algorithm for pose estimation in elderly care systems offers significant benefits in enhancing fall detection capabilities. The real-time processing and accuracy of the YOLO algorithm, combined with the detailed movement analysis provided by pose estimation, create a powerful and reliable monitoring solution.

CUDA Python was specifically chosen to be used due to its ability to significantly improve computational efficiency. The GPU, a specialized processor designed to accelerate graphics rendering, features a highly parallel structure that is ideal for executing multiple computations simultaneously.

This integrative approach not only ensures prompt and accurate detection of falls but also supports care and personalized monitoring of the elderly. By integrating an alert system through Telegram, we can ensure a quick reaction in case of incidents, thus improving the quality of life of those who need constant surveillance. The flexible architecture and high performance of YOLO for pose estimation make this solution particularly valuable in the fields of health monitoring and care for the elderly. Also, our system adheres to privacy standards, guaranteeing that processed information is not saved. This data protection measure is essential to ensure the security and privacy of users, avoiding unauthorized storage and access to sensitive data.

In conclusion, the application of the YOLO algorithm for pose estimation in elderly care represents significant advancements. This approach not only improves the efficiency and effectiveness of care but also ensures that elderly individuals receive the attention and support they need in a dignified and respectful manner. As technology continues to evolve, the integration of such cutting-edge solutions will be essential in addressing the growing demands of elderly care.

5.1. Limitations

The implementation of a fall detection system utilizing YOLO algorithms with pose estimation presents significant challenges due to its demand for substantial computational power required for real-time processing. The high-quality INSTA360 GO3 camera, the high-performance GPUs, and the additional hardware are cost-effective but required to achieve the desired speed and accuracy of the system. Addressing these constraints will be important for broader deployment and efficient system maintenance.

Currently, the system differentiates a fall from someone lying down based on the sudden and abrupt movement typically associated with a fall. However, despite the high accuracy of YOLO, no system is infallible, and false positives and false negatives can still occur. False positives may lead to unnecessary stress and resource utilization, while false negatives could result in missed emergencies. Expanding the system's ability to cover edge cases will be essential to ensuring reliable fall detection in diverse environments. Continuous refinement of the algorithms and rigorous testing are necessary to minimize these issues, as the dynamic nature of fall scenarios and varying environmental conditions require ongoing adjustments to maintain high detection accuracy and overall system reliability.

The system was not evaluated (1) under the full range of lighting variations found in real-world environments (for example, mixed lighting, rapid lighting changes, or areas with shadows may impact the system's accuracy); (2) in real-world environments where parts of a person's body may be hidden by objects, furniture, or other people, which can make it difficult for the system to detect a fall accurately; (3) taking into account the full diversity of physical abilities of the individuals being monitored (differences in movement patterns, walking speed, and the manner in which individuals fall can all influence detection accuracy); or (4) in scenarios where a person is performing exercises on a mat on the floor. The system has not been tested to determine if such activities may trigger false alarm notifications, as movements during exercise could resemble a fall.

5.2. Future Directions

Looking ahead, the application of the YOLO algorithm for pose estimation in fall detection for elderly care holds tremendous promise. These advancements could significantly enhance safety monitoring and health management for seniors. These approaches can be fine-tuned to accommodate the specific needs and behaviors of individual elderly persons. For instance, the system can learn and adapt to the unique movement patterns of each person, improving the accuracy of fall detection and reducing the likelihood of false alarms tailored to the individual's specific conditions and living environment. The use of wearable devices (such as accelerometers and gyroscopes) can provide additional context and verification, reducing false positives and negatives. Combining video-based fall detection with data from wearable devices will enhance accuracy and reliability. Continued advancements in machine learning and AI can lead to more sophisticated algorithms that improve detection accuracy and adaptability. Research into new models that can better handle diverse environments and conditions will be beneficial.

Enabling proactive and customizable alerts to caregivers or emergency services in case of emergencies could be beneficial in ensuring timely assistance. Moreover, the continuous refinement and adaptation of these algorithms could lead to personalized care solutions, promoting independence and quality of life among the elderly population. As research and development in this field progress, ethical considerations regarding privacy, consent, and data security will remain paramount to ensure the responsible deployment and adoption of these solutions in elder care settings.

The current testing did not specifically address complex backgrounds (e.g., cluttered environments with multiple objects or people), but future work could investigate how the system performs in such conditions. In typical testing scenarios, the backgrounds are relatively simple and do not interfere significantly with detection. However, environments with a high level of visual complexity (such as crowded spaces or areas with numerous objects) could present challenges for the system. Further testing will be necessary to assess how well the system adapts to such conditions, particularly in distinguishing the fall event from environmental clutter.

Further testing is needed to evaluate the system's performance under different lighting conditions, ensuring it can adapt to dynamic environments without sacrificing fall detection accuracy. Future research should also examine how the system handles occlusions, particularly in crowded or cluttered spaces. Additionally, studies should include a broader range of participants to assess the system's effectiveness across a more diverse population, such as elderly individuals with mobility impairments or those who may fall in different ways due to medical conditions. Another important consideration is testing the system in scenarios where a person is exercising on a floor mat, as movements during exercise may resemble a fall and potentially trigger false alarms.

Future work will also involve extensive real-world validation in healthcare settings to better understand the system's performance and user acceptance. Collaborations with healthcare providers will be pursued to enable assessment in diverse and dynamic environments, enhancing reliability and ensuring practical applicability.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/fi16120472/s1, Supplementary Materials S1—The detailed workings of the advanced fall detection system. Reference [66] are cited in the supplementary materials.

Author Contributions: Conceptualization, methodology, software, investigation, resources, formal analysis, and validation, E.T. (Eugenia Tîrziu), A.-M.V., A.A. and E.T. (Eleonora Tudora); data curation, E.T. (Eugenia Tîrziu) and A.-M.V.; writing—original draft preparation, E.T. (Eugenia Tîrziu), A.-M.V., A.A. and E.T. (Eleonora Tudora); writing—review and editing, A.A., A.-M.V. and E.T. (Eleonora Tudora); supervision, project administration, and funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Romanian Ministry of Research, Innovation, and Digitalization, grant number 13N/2023 (PN 23 38 05 01). The APC was funded by MDPI.

Data Availability Statement: The data presented in this study are openly available in GitHub at https://github.com/WongKinYiu/yolov7?tab=readme-ov-file (accessed on 16 December 2024).

Acknowledgments: The authors gratefully acknowledge the contribution of the Romanian Ministry of Research, Innovation, and Digitalization for the development of the NeuroPredict Platform inside the project "Advanced Artificial Intelligence Techniques in Science and Applications" (contract No. 13N/2023 (PN 23 38 05 01)) for the period 2023–2026.

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

References

1. United Nations. *World Population Ageing 2019: Highlights*; United Nations: New York, NY, USA, 2019.
2. World Health Organization. Falls. Available online: https://www.who.int/en/news-room/fact-sheets/detail/falls (accessed on 26 June 2024).
3. Yu, Z.; Liu, J.; Yang, M.; Cheng, Y.; Hu, J.; Li, X. An Elderly Fall Detection Method Based on Federated Learning and Extreme Learning Machine (Fed-ELM). *IEEE Access* **2022**, *10*, 130816–130824. [CrossRef]
4. Băjenaru, L.; Marinescu, I.A.; Dobre, C.; Drăghici, R.; Herghelegiu, A.M.; Rusu, A. Identifying the Needs of Older People for Personalized Assistive Solutions in Romanian Healthcare System. *Stud. Inform. Control* **2020**, *29*, 363–372. [CrossRef]
5. Nguyen, H.H. A Comprehensive Approach to Secure and Effective Fall Detection in IOT Healthcare Systems. Master's Thesis, Auckland University of Technology, Auckland, New Zealand, 2023.
6. Gomes, M.E.N.; Macêdo, D.; Zanchettin, C.; Gomes de-Mattos-Neto, P.S.; Oliveira, A. Multi-human Fall Detection and Localization in Videos. *Comp. Vision Image Underst.* **2022**, *220*, 103442. [CrossRef]
7. Newaz, N.T.; Hanada, E. The Methods of Fall Detection: A Literature Review. *Sensors* **2023**, *23*, 5212. [CrossRef] [PubMed]
8. Apostol, M.; Duțescu, R.-A. Remote fall detection system for elderly people using non-invasive technologies. *Rom. J. Inform. Techn. Autom. Control* **2023**, *33*, 33–42. [CrossRef]
9. Usmani, S.; Saboor, A.; Haris, M.; Khan, M.A.; Park, H. Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors* **2021**, *21*, 5134. [CrossRef] [PubMed]
10. Thakur, N.; Han, C.Y. A Simplistic and Cost-Effective Design for Real-World Development of an Ambient Assisted Living System for Fall Detection and Indoor Localization: Proof-of-Concept. *Information* **2022**, *13*, 363. [CrossRef]
11. Wang, X.; Ellul, J.; Azzopardi, G. Elderly Fall Detection Systems: A Literature Survey. *Front. Robot. AI* **2020**, *23*, 71. [CrossRef] [PubMed]
12. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475. [CrossRef]
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016. ECCV 2016. Lecture Notes in Computer Science (LNCS)*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Germany, 2016; Volume 9905. [CrossRef]
14. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
15. Aboyomi, D.; Daniel, C. A Comparative Analysis of Modern Object Detection Algorithms: YOLO vs. SSD vs. Faster R-CNN. *Inf. Tech. Eng. J. (ITEJ)* **2023**, *8*, 96–106. [CrossRef]
16. Tan, L.; Huangfu, T.; Wu, L.; Chen, W. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. *BMC Med. Inf. Decis. Mak.* **2021**, *21*, 324. [CrossRef] [PubMed]
17. Kamble, K.P.; Sontakke, S.S.; Donadkar, H.; Poshattiwar, R.; Ananth, A. Fall Alert: A Novel Approach to Detect Fall Using Base as a YOLO Object Detection. In *Advanced Machine Learning Technologies and Applications. AMLTA 2020. Advances in Intelligent Systems and Computing*; Hassanien, A., Bhatnagar, R., Darwish, A., Eds.; Springer: Singapore, 2021; Volume 1141. [CrossRef]
18. Gao, P. Development of YOLO-based Model for Fall Detection in IoT Smart Home Applications. *Int. J. A. Comp. Sci. Applic. (IJACSA)* **2023**, *14*, 1118–1125. [CrossRef]
19. Tuan Rashid, T.M.N.B.; Mohd Fadzil, L.; Mohd, A.H.O. 3D Bounding Box Estimation Using Deep Learning and Geometry Based on Yolov7 Output on Single-Board Computer. *SSRG Int. J. Electr. Electron. Eng.* **2024**, *11*, 77–84. [CrossRef]
20. Long, K.Z.; Haron, H.; Ibrahim, M.; Eri, Z.D. An Image-based Fall Detection System using You Only Look Once (YOLO) Algorithm to Monitor Elders' Fall Events. In Proceedings of the Knowledge Management International Conference (KMICe), Changlun, Malaysia, 1 February 2021; Universiti Utara Malaysia: Changlun, Malaysia, 2021.
21. Li, Y.; Zhang, W.; Ma, H.; Chen, F.; Sun, D.; Cheng, X.; Zheng, Y.; Wang, Z.W.; Cai, S. Posture Detection Based on Kinect and YOLO Technologies. In Proceedings of the 2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI), Hangzhou, China, 1–3 December 2023; pp. 205–209. [CrossRef]
22. Lee, J.; Hwang, K.-il. YOLO with adaptive frame control for real-time object detection applications. *Multimed. Tools. Appl.* **2022**, *81*, 36375–36396. [CrossRef]

23. Waghchaware, S.; Joshi, R.D. Estimation of Human Fall Detectionusing Body Pose Rule based Algorithm for Video Surveillance. In Proceedings of the 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 5–7 April 2024; pp. 1–5. [CrossRef]
24. Vyas, H.; Sharma, S.; Senghani, H.; Rathod, A.; Vasant, A. Accident Prone System using YOLO. *Int. J. Sci. Res. Sci. Eng. Tech. (IJSRSET)* **2023**, *10*, 09–16. [CrossRef]
25. Qin, Y.; Miao, W.; Qian, C. A High-Precision Fall Detection Model Based on Dynamic Convolution in Complex Scenes. *Electronics* **2024**, *13*, 1141. [CrossRef]
26. Jiang, Y.; Hang, R.; Wu, Y.; Huang, W.; Pan, X.; Tao, Z. Design and Development of Posture Detection System Based on YOLO—OpenPose. In Proceedings of the 2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Harbin, China, 29–31 July 2023; pp. 1–7. [CrossRef]
27. Sohal, R.S.; Singh, M.P.; Singh, K. Pose Estimation for Human Activity Recognition Using Deep Learning on Video Data. *Proc. Int. J. Sci. Tech. (PIJST)* **2024**, *1*, 13–21. [CrossRef]
28. Jia, Z.; Zhang, H.; Jia, Y.; Zheng, Y.; Li, D.; Jia, S. Human fall detection scheme based on YOLO visual recognition and embedded ARM architecture. In Proceedings of the Second International Conference on Algorithms, Microchips, and Network Applications (AMNA 2023), Zhengzhou, China, 8 May 2023; Volume 12635, p. 1263514. [CrossRef]
29. Phatangare, S.; Kate, S.; Khandelwal, D.; Khandetod, A.; Kharade, A. Real Time Human Activity Detection using YOLOv7. In Proceedings of the 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 11–13 October 2023; pp. 1069–1076. [CrossRef]
30. Haider, S.; Mughal, A.; Khan, R. DeepPoseNet: A Comprehensive Study on Human Pose Estimation with Deep Learning Technique. *Int. J. Artif. Int. Math. Sci.* **2024**, *2*, 1–10. [CrossRef]
31. Zheng, Y.; Zhang, S. Research on fall detection based on improved human posture estimation algorithm. *Instrumentation* **2021**, *8*, 18–33.
32. Calvache, D.A.; Bernal, H.A.; Guarín, J.F.; Aguía, K.; Orjuela-Cañón, A.D.; Perdomo, O.J. Automatic estimation of pose and falls in videos using computer vision model. In Proceedings of the 16th International Symposium on Medical Information Processing and Analysis, Lima, Peru, 3 November 2020; Volume 11583, p. 115830W. [CrossRef]
33. Gong, M.; Liu, R.; Asari, V.K. YOLO-based GNN for multi-person pose estimation. In Proceedings of the Pattern Recognition and Prediction XXXV, National Harbor, MD, USA, 7 June 2024; Volume 13040, p. 130400F. [CrossRef]
34. CUDA. Available online: https://developer.nvidia.com/cuda-zone#:~:text=CUDA%C2%AE%20is%20a%20parallel,harnessing%20the%20power%20of%20GPUs (accessed on 26 June 2024).
35. CUDA Python. Available online: https://developer.nvidia.com/cuda-python (accessed on 26 June 2024).
36. Kavitha, A.; Hemalatha, B.; Abishek, K.; Harigokul, R. Fall Detection of Elderly Using YOLO. In *ICT Systems and Sustainability. Lecture Notes in Networks and Systems*; Tuba, M., Akashe, S., Joshi, A., Eds.; Springer: Singapore, 2023; Volume 516, pp. 113–121. [CrossRef]
37. Nguyen, H.; Mai, T.; Nguyen, M. A Holistic Approach to Elderly Safety: Sensor Fusion, Fall Detection, and Privacy-Preserving Techniques. In Proceedings of the Image and Video Technology: 11th Pacific-Rim Symposium, PSIVT 2023, Auckland, New Zealand, 22–24 November 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 380–393. [CrossRef]
38. Luo, Z.; Jia, S.; Niu, H.; Zhao, Y.; Zeng, X.; Dong, G. Elderly Fall Detection Algorithm Based on Improved YOLOv5s. *Inf Tech Cont.* **2024**, *53*, 601–618. [CrossRef]
39. Pooja, K.V.; Pahuja, S.K.; Veer, K. IoT and Machine Learning Algorithms for Fall Detection. In *Wearable/Personal Monitoring Devices Present to Future*; Gargiulo, G.D., Naik, G.R., Eds.; Springer: Singapore, 2022. [CrossRef]
40. Stampfler, T.; Elgendi, M.; Fletcher, R.R.; Menon, C. Fall detection using accelerometer-based smartphones: Where do we go from here? *Front. Public Health* **2022**, *10*, 996021. [CrossRef]
41. Kolobe, T.; Tu, C.; Owolawi, P. A Review on Fall Detection in Smart Home for Elderly and Disabled People. *J. Adv. Comput. Intell. Int. Inform.* **2022**, *26*, 747–757. [CrossRef]
42. Rajaei, H.; Upendra, S. Intelligent Fall Detection Monitoring System. In Proceedings of the Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 24–27 July 2023; pp. 1721–1727. [CrossRef]
43. Zerrouki, N.; Harrou, F.; Sun, Y.; Djafer, A.Z.A.; Amrane, H. A Survey on Recent Advances in Fall Detection Systems Using Machine Learning Formalisms. In Proceedings of the 2022 7th International Conference on Frontiers of Signal Processing (ICFSP), Paris, France, 7–9 September 2022; pp. 35–39. [CrossRef]
44. Sykes, E.R. An Analysis of Current Fall Detection Systems and the Role of Smart Devices and Machine Learning in Future Systems. In *Advances in Information and Communication*; FICC 2023. Lecture Notes in Networks and Systems; Arai, K., Ed.; Springer: Cham, Germany, 2023; Volume 652. [CrossRef]
45. Shreya, L.L. Fall Detection for Elderly People Using Machine Learning. *Int. J. Res. App. Sci. Eng. Tech.* **2023**, *11(XII)*, 798–802. [CrossRef]
46. Qiu, Y.; Meng, J.; Li, B. Automated Falls Detection Using Visual Anomaly Detection and Pose-based Approaches: Experimental Review and Evaluation. *J. Biomed. Res. Environ. Sci.* **2024**, *5*, 55–63. [CrossRef]
47. Basu, I.; Ray, S.; Sarkar, S.; Saha, S.; Panda, S. Advancements in Fall Detection Technologies: A Comprehensive Review and Innovative Solutions. In Proceedings of the 2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), Kolkata, India, 18–20 December 2023; pp. 1–3. [CrossRef]

48. Al-Risi, A.A.N.; Albadi, S.S.M.; Almaamari, S.H.S.; Samad, S.R.A.; Ganesan, P. Automated Fall Detection for Disabled Individuals Using Mobile Phone Sensors and Machine Learning: A Survey. *Int. J. Data Inf. Int. Comp.* **2024**, *3*, 27–33. [CrossRef]
49. Alharbi, H.A.; Alharbi, K.K.; Hassan, C.A.U. Enhancing Elderly Fall Detection through IoT-Enabled Smart Flooring and AI for Independent Living Sustainability. *Sustainability* **2023**, *15*, 15695. [CrossRef]
50. Yin, Z.; Yan, J.; Fang, S.; Wang, D.; Han, D. User acceptance of wearable intelligent medical devices through a modified unified theory of acceptance and use of technology. *Ann. Transl. Med.* **2022**, *10*, 629. [CrossRef]
51. Liu, J.; Tan, R.; Han, G.; Sun, N.; Kwong, S. Privacy-Preserving In-Home Fall Detection Using Visual Shielding Sensing and Private Information-Embedding. *IEEE Trans. Mul.* **2021**, *23*, 3684–3699. [CrossRef]
52. OpenCV. Available online: https://opencv.org/about/ (accessed on 26 June 2024).
53. PyTorch Documentation. Available online: https://pytorch.org/docs/stable/community/design.html (accessed on 26 June 2024).
54. Torchvision. Available online: https://pytorch.org/vision/stable/index.html (accessed on 26 June 2024).
55. NumPy Documentation. Available online: https://numpy.org/doc/stable/ (accessed on 26 June 2024).
56. WongKinYiu-Yolov7 Dataset. Available online: https://github.com/WongKinYiu/yolov7?tab=readme-ov-file (accessed on 25 September 2024).
57. Building a Telegram Bot with Python: A Step-by-Step Guide. Available online: https://medium.com/@rubyabdullah14/building-a-telegram-bot-with-python-a-step-by-step-guide-5ca305bea6c0 (accessed on 26 June 2024).
58. Chen, W.; Jiang, Z.; Guo, H.; Ni, X. Fall Detection Based on Key Points of Human-Skeleton Using OpenPose. *Symmetry* **2020**, *12*, 744. [CrossRef]
59. Free Person Falling Videos from Pexels. Available online: https://www.pexels.com/ (accessed on 25 September 2024).
60. Free Person Falling Videos from Pixabay. Available online: https://pixabay.com/ro/ (accessed on 25 September 2024).
61. Free Person Falling Stock Videos from Mixkit. Available online: https://mixkit.co/ (accessed on 25 September 2024).
62. Le2i Fall Dataset. Available online: https://www.kaggle.com/datasets/tuyenldvn/falldataset-imvia (accessed on 25 September 2024).
63. Chamle, M.; Gunale, K.G.; Warhade, K.K. Automated unusual event detection in video surveillance. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016.
64. Alaoui, A.Y.; El Hassouny, A.; Thami, R.O.H.; Tairi, H. Human Fall Detection Using von Mises Distribution and Motion Vectors of Interest Points. *Assoc. Comput. Mach.* **2017**, *82*, 5. [CrossRef]
65. Poonsri, A.; Chiracharit, W. Improvement of fall detection using consecutive-frame voting. In Proceedings of the International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018. [CrossRef]
66. Telegram. Available online: https://core.telegram.org/ (accessed on 26 June 2024).