ASEAN
Journal of Scientific and Technological Reports
Online ISSN:2773-8752

Research article

An IoT-Based Real-Time Human Fall Detection and Notification System with Instance Segmentation Deep Learning

Wiwat Su-hren[1], Tanawat Srirugsa[1, 4], Saowanee Singsarothai[1], Supachai Kaewpoung[1,4] and Tawat Chuchit[1, 4*]

[1] Faculty of Engineering, Thaksin University, Phatthalung 93210, Thailand
[4] Center of Excellence for Agricultural Innovation and Bioproducts of Thaksin University, Thaksin University, Phatthalung, 93210, Thailand
* Correspondence: tawat.c@tsu.ac.th; (T. Chuchit)

Citation:
Suhren, W., Srirugsa, T., Singsarothai, S., Kaewpoung, S., Chuchit, T., An IoT-based real-time human fall detection and notification system with instance segmentation eeep learning. *ASEAN J. Sci. Tech. Report.* **2026**, *29*(3), e260412. https://doi.org /10.55164/ajstr. v29i3.260412.

Article history:
Received: July 19, 2025
Revised: January 6, 2026
Accepted: January 11, 2026
Available online: February 28, 2026

Abstract: This research presents a person fall detection system based on deep learning using the Instance Image Segmentation technique with the YOLOv11-seg model, which achieves high speed and accuracy in object detection. The developed system aims to individually detect a person's posture in an image, enabling accurate analysis of their falling posture characteristics. And there is a notification in the application to the administrator or relevant person within 10 seconds, allowing quick help. Using a dataset of 10,169 images, 8622 for training, 994 for inspection, and 553 for testing (with a ratio of 85:10:5 for training, inspection, and testing). The system performs impressively by using all 3 YOLOv11-seg models: YOLOv11s-seg, YOLOv11n-seg, and YOLOv11m-seg. The training dataset showed excellent performance with a YOLOv11s-seg model, achieving a precision of 0.966, a recall of 0.910, and an F1 Score of 0.88. The results show that the developed system can detect falls and issue real-time alerts via an IoT-based framework. It improves safety and reduces the risk for the elderly or patients at risk of falling.

Keywords: Instance segmentation; deep learning; human fall detection system; YOLO; IoT-based framework

1. Introduction

Fall-related injuries are a major problem affecting the quality of life of the elderly and high-risk patients. According to the World Health Organization (WHO), falls are the leading cause of severe injuries and a major factor in premature deaths among the elderly [1], with the trend of an increasing elderly population around the world. This problem has attracted increased attention in the research community, leading to the development of systems that can detect falls and provide timely assistance. Falling detection with computer vision and artificial intelligence (AI) technology has become popular because it is a non-intrusive method and can be widely applied in the real world. Especially when deep learning technology is used to help analyze human movement behavior. This significantly increases the detection accuracy. One technique that has gained attention is Instance Segmentation, which can segment and classify individual objects in an image at the pixel level [2]. It enables more effective detection of falling behavior than traditional methods based on Object Detection or Pose Estimation. In some cases, there is overlap among individuals or a complex environment. Past research has shown that using pose skeletons rather than boundaries can effectively improve the efficiency of individual discrimination [3]. In addition, approaches based on image sequence analysis with 3D CNNs and temporal models such as LSTMs have become increasingly popular [4,5]. These techniques can temporarily extract movement data to help better distinguish falling behavior from normal movement. Meanwhile, Techniques that use skeleton-based detection in conjunction with temporal networks, such as Two-stage Temporal Convolutional Networks (TCN) and models that integrate OpenPose with high-performance networks, have been developed to meet the needs of real-time detection [6,7]. However, past research has identified an important trend: integrating deep learning techniques with multi-perspective visual data and temporal motion analysis, which is the main direction of fall detection technology today. [8].

The main goal of this research is to create a system that can be used in a variety of environments. It supports real-time processing and helps enhance the safety of the elderly and patients at high risk of falling in their daily lives. This research aims to present a new approach to applying Deep Learning and Instance Segmentation in healthcare and accident surveillance in the future.

2. Materials and Methods

2.1 Instance segmentation

In the past decade, the rapid advancement of Artificial Intelligence (AI) has driven significant progress in computer vision, particularly in image understanding and object segmentation. Among various techniques, instance segmentation has emerged as one of the most important and widely adopted approaches, integrating the strengths of object detection and semantic segmentation. While object detection provides coarse localization through bounding boxes and semantic segmentation assigns class labels to pixels without distinguishing individual objects, instance segmentation enables the precise localization and delineation of each object instance at the pixel level. This capability is particularly important for applications that require accurate spatial understanding, such as human posture analysis and fall detection. Instance segmentation techniques were initially popularized by the Mask R-CNN framework, which extends the Faster R-CNN architecture by introducing a parallel branch for pixel-level mask prediction, as illustrated in Figure 1. In Mask R-CNN, region proposals generated by the Region Proposal Network (RPN) are processed simultaneously by a classification branch and a mask branch. A fixed-size feature representation is passed through a SoftMax function to produce class probabilities, while a Fully Connected Layer followed by deconvolution layers generates a binary segmentation mask corresponding to the detected object at its original spatial resolution. Although this design achieves high segmentation accuracy, it incurs relatively high computational complexity, which limits its suitability for real-time and edge-based applications [9]. To overcome the limitations of two-stage architectures, one-stage detection models, such as YOLO (You Only Look Once), have been extensively developed. YOLO is well known for its real-time detection capability, as it performs object localization and classification in a single forward pass. The YOLO family has evolved continuously, and the latest variant, YOLOv11-seg, extends the original detection framework to support instance segmentation while maintaining high inference speed. This characteristic makes YOLOv11-seg particularly suitable for applications that require real-time processing and low-latency deployment.

YOLOv11-seg adopts a one-stage architecture composed of three main components: a backbone, a neck, and a head. The backbone is responsible for feature extraction and is built on an improved Cross Stage Partial (CSP) bottleneck structure that enhances feature reuse while reducing computational redundancy. The neck incorporates modules such as Spatial Pyramid Pooling–Fast (SPPF) to aggregate multi-scale contextual information, enabling robust object detection across varying sizes under diverse scene conditions. In addition, integrating a Convolutional block with Parallel Spatial Attention (C2PSA) enables the network to focus more effectively on informative spatial regions within the input image. The instance segmentation capability of YOLOv11-seg is realized through a Protonet-based mask generation module integrated into the detection head. This module generates a set of shared prototype masks from high-level feature maps. For each detected object, the network predicts a set of mask coefficients that linearly combine these prototypes to produce an instance-specific segmentation mask. This mechanism is conceptually similar to that employed in the YOLACT model; however, it is optimized to integrate seamlessly with the YOLO architecture, thereby enabling faster inference and improved suitability for real-time applications. Deconvolution and up-sampling operations are subsequently applied to align the generated masks with the original image resolution, ensuring accurate object

boundaries and a faithful representation of object shapes [10,11]. By leveraging instance segmentation with YOLOv11-seg, the proposed system can obtain fine-grained pixel-level representations of human posture, which are essential for reliably distinguishing between standing and falling states. This detailed spatial information enhances robustness in challenging scenarios, such as partial occlusions, overlapping individuals, and varying illumination conditions. Consequently, YOLOv11-seg provides an effective and computationally efficient foundation for real-time human fall detection systems deployed on edge and IoT-based platforms.

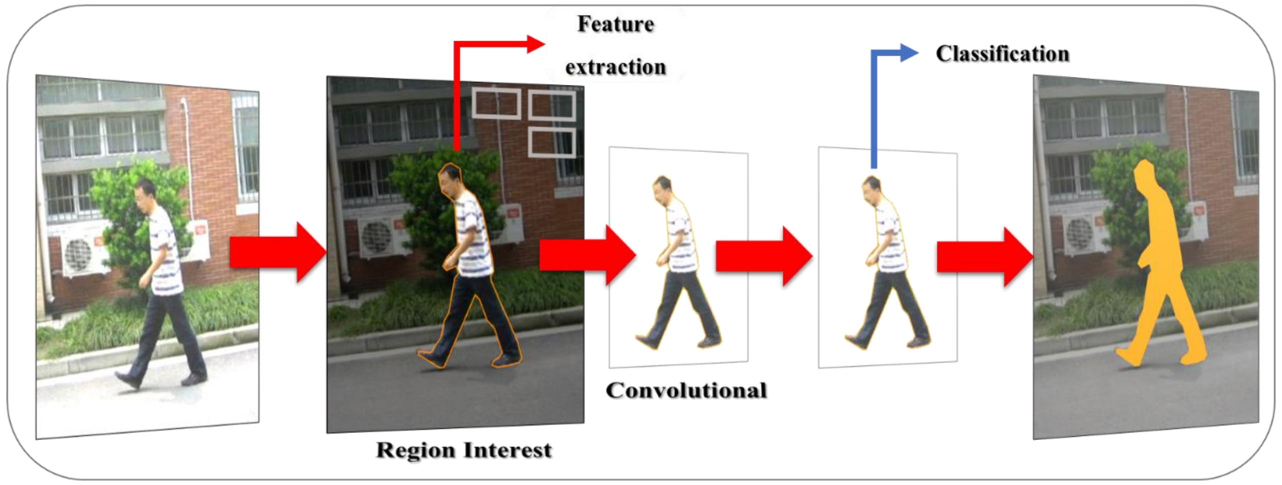

Figure 1. Principle of Mask R-CNN

2.2 ROBOFLOW

Dataset preparation is one of the most important steps in detecting and segmenting images. ROBOFLOW is a cloud-based platform designed to facilitate the preparation of visual data for computer vision tasks, especially for AI training in object detection, Image Segmentation, and Image Classification. In addition, ROBOFLOW offers data management capabilities, including data import into the platform, image annotation, image augmentation, and the creation of datasets suitable for model training in formats such as YOLO, COCO, Pascal VOC, or TFRecord. This makes ROBOFLOW an important tool that effectively reduces the technical workload and increases the speed of AI model development [12]. ROBOFLOW also supports end-to-end model training on the cloud. Users can quickly bring the datasets obtained from annotation into the training process with popular models such as YOLOv5, YOLOv8, Detectron2, or Segment Anything, as shown in Figure 2. It also supports releasing models via API immediately after annotation, which is ideal for researchers, developers, and entrepreneurs who want to experiment with it or use it in systems that require image processing, such as detecting defects in production lines or counting customers in stores. This has made it very popular in the AI industry.

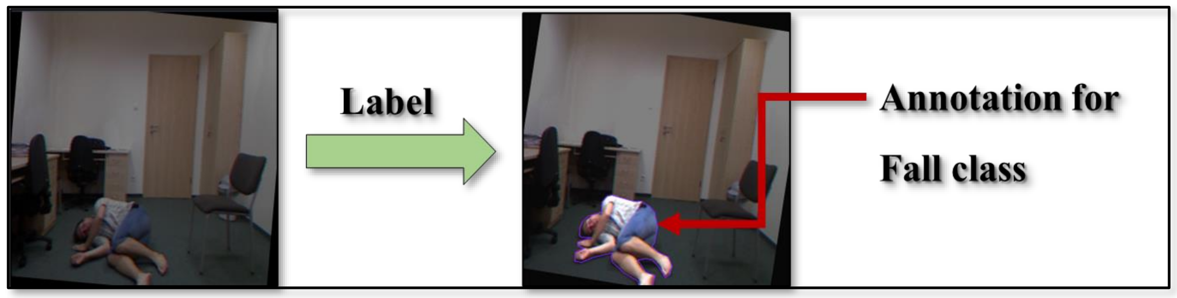

Figure 2. Annotation process.

2.3 Model evaluation

To evaluate the model, Precision (P), Recall (R), mPA@0.5, and F1-scores are used to assess its performance. Three YOLOv11-seg models were used in the test: YOLOv11s-seg, YOLOv11n-seg, and YOLOv11m-seg [13]. The Precision (P) value represents the accuracy of the prediction, which is calculated by the number of true positives (TP) divided by the sum of the number of true positives (TP) and false positives (FP), as shown in equation 1. The Recall (R) value is used to evaluate the performance of a model that affects an object of interest. It is calculated as the number of true positives (TP) divided by the sum of true positives (TP) and false negatives (FN), as shown in equation 2. The mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) represents the overall detection and segmentation performance across all classes by jointly considering localization and classification accuracy. As shown in equation 3. [14, 15]

$$P = TP / (TP + FP) \tag{1}$$

$$R = TP / (TP + FN) \tag{2}$$

$$\text{mPA@0.5} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{3}$$

where N is the number of classes, and AP denotes the Average Precision of class iii, obtained from the area under the Precision–Recall curve at an IoU threshold of 0.5. The mAP@0.5 metric balances detection robustness and localization tolerance and is therefore commonly adopted in real-time and edge-based detection systems. More stringent metrics, such as mAP@0.5:0.95, were not employed in this study because extremely strict localization accuracy is not essential for practical fall detection decisions. F1-Scores combines recall and precision to find the harmonic mean to solve the imbalanced classification problem, as shown in equation 4. [16,17]

$$\text{F1-Scores} = 2 \times P \times R / (P + R) \tag{4}$$

Together, these evaluation metrics provide a comprehensive assessment of the proposed system in terms of detection precision, robustness, and suitability for real-time deployment in IoT-based healthcare monitoring applications.

2.4 Integration with IoT and Real-Time Notification

To enhance the practical usability of the proposed fall detection system, the system integrates Internet of Things (IoT) technology with a real-time notification mechanism. After the YOLOv11-seg model detects a fall and confirms that the individual remains motionless for more than 10 seconds, an alert is automatically transmitted to a caregiver or designated recipient. All detection and notification processes are executed locally on an edge device, such as a Jetson Nano, a personal computer, or a Raspberry Pi, without relying on cloud-based inference, thereby reducing latency and preserving data privacy [18]. Alert notifications are delivered via LINE Notify, a messaging API widely used in Thailand, Japan, and Taiwan. The system transmits messages containing the detection timestamp, fall status, and optionally a snapshot image of the incident to the recipient's smartphone, enabling timely awareness and rapid response [19]. The end-to-end notification latency consists of three components: frame-level detection and instance segmentation, posture confirmation, and message transmission. The YOLOv11-seg model operates in real time with an average inference time of less than 500 ms per frame. A fixed 10-second immobility threshold is applied to reduce false alarms, after which the alert is sent via a secure HTTPS POST request. Under normal network conditions, message delivery latency ranges between 1 and 3 seconds [20–22]. Consequently, the total end-to-end notification latency is consistently within approximately 12 seconds, confirming the suitability of the proposed system for real-time IoT-based fall detection applications.

3. Experiment

In this study, experiments were conducted to evaluate the performance of the proposed instance segmentation–based human fall detection system using the YOLOv11-seg model. The dataset used for detection and segmentation was collected from two primary sources. The first source consisted of images captured within the Thaksin University, Phatthalung Campus, representing real-world indoor and outdoor environments with varying illumination conditions, backgrounds, and camera viewpoints. The second source comprised curated images obtained from the Roboflow platform, selected to enhance posture diversity and scene variation. The dataset was categorized into two posture classes: Stand and Fall. The raw dataset contained 1,778 images in the Stand class and 3,878 images in the Fall class, as summarized in Table 1. To improve model robustness and generalization, data augmentation techniques were applied, including brightness and exposure adjustments. These augmentations were designed to simulate lighting variations commonly encountered in practical deployment scenarios, such as indoor environments and low-light conditions. After augmentation, the dataset was expanded to 10,169 images. The augmented dataset was divided into three subsets following an 85:10:5 ratio: 8,622 images for training, 994 images for validation, and 553 images for testing. An overview of the dataset composition is presented in Table 2, while the software tools and hardware environment used for model training and evaluation are summarized in Table 3. This data partitioning strategy was selected to ensure sufficient training samples while maintaining independent validation and testing sets for unbiased performance assessment. All experiments were conducted on a workstation equipped with an Intel Core i5-10300H CPU, an NVIDIA GTX 2060 GPU, and 16 GB of RAM. The models were implemented using Python 3.10 with CUDA 11.8 support. Due to hardware constraints, experiments were performed on a single-GPU platform. Although this configuration limits large-scale scalability evaluation, it is representative of edge-level deployment environments targeted by the proposed system, where computational resources are typically constrained.

Table 1. Data of Raw images

Class	Number of Raw images
Stand	1,778
Fall	3,878

Table 2. An overview of the dataset

Title	Description
Number of Classes	2
Total Number of Input Images	10,169
Training Images	8,622
Validation Images	994
Testing Images	553
Brightness	±13%
Exposure	±15%

Table 3. Environments used to train the model

Operating environment	Version
Operating System	Window 11
Language	Python 3.10
CUDA	11.8
GPU	GTX2060
CPU	Intel(R) Core(TM) i5-10300H

Table 4 summarizes the key training hyperparameters used for all YOLOv11-seg model variants in this study. To ensure a fair and consistent comparison, a unified training configuration was applied across all models. The input image resolution was fixed at 640 × 640 pixels, and the batch size was set to 16 samples per iteration. All models were trained for 100 epochs using the Adam optimizer with an initial learning rate of 0.01, with cosine learning rate decay to promote stable convergence. Standard YOLOv11-seg loss components, including bounding box loss, classification loss, and mask loss, were employed during training. Data augmentation techniques, such as random scaling and color adjustment, were enabled to improve generalization under varying illumination and environmental conditions. By maintaining identical hyperparameter settings for all YOLOv11-seg variants, observed performance differences can be attributed primarily to architectural variations rather than training-related bias.

Table 4. Training Hyperparameters

Hyperparameter	Value
Input image size	640 × 640
Batch size	16
Optimizer	Adam
Initial learning rate	0.01
Number of epochs	100
Loss functions	Box, Class, Mask loss

In the design of the fall detection system in this study, the working principle is clearly defined in stages. This is illustrated in Figure 3, which provides an overview of the process from inception to notification. In the design of the fall detection system in this study, the working principle is clearly defined in stages. This is illustrated in Figure 3, which provides an overview of the process from inception to notification. When the system starts, a pre-trained YOLOv11-seg model is loaded and trained on a dataset divided into "standing" and "falling" classes. The model has been adjusted using various parameters and weights from the training process to accurately distinguish a person's behavior in the image at the pixel level using the Instance Segmentation technique. After the image or video is sent to the system, the model continuously analyzes each image frame to detect a person's posture in the scene. If the system detects that a person is in a posture consistent with the nature of the fall, the person is in a position that corresponds to that nature. The system will begin the automatic timer process. If within 10 seconds after the fall. The person remains in the same position, without any movements indicating recovery, such as supporting or getting up. The system concludes that it is a fall case requiring assistance and immediately sends a notification to the administrator or relevant person via a defined system, such as an application, an IoT device notification, or a messaging system. The entire sequence of steps for this system is designed to be easy to apply in a real environment and to offer a high degree of flexibility in deployment across different devices. Image Analysis Decisions and alerts are collected and displayed in a flowchart format, as shown in Figure 4, to provide a clear overview of all mechanisms.

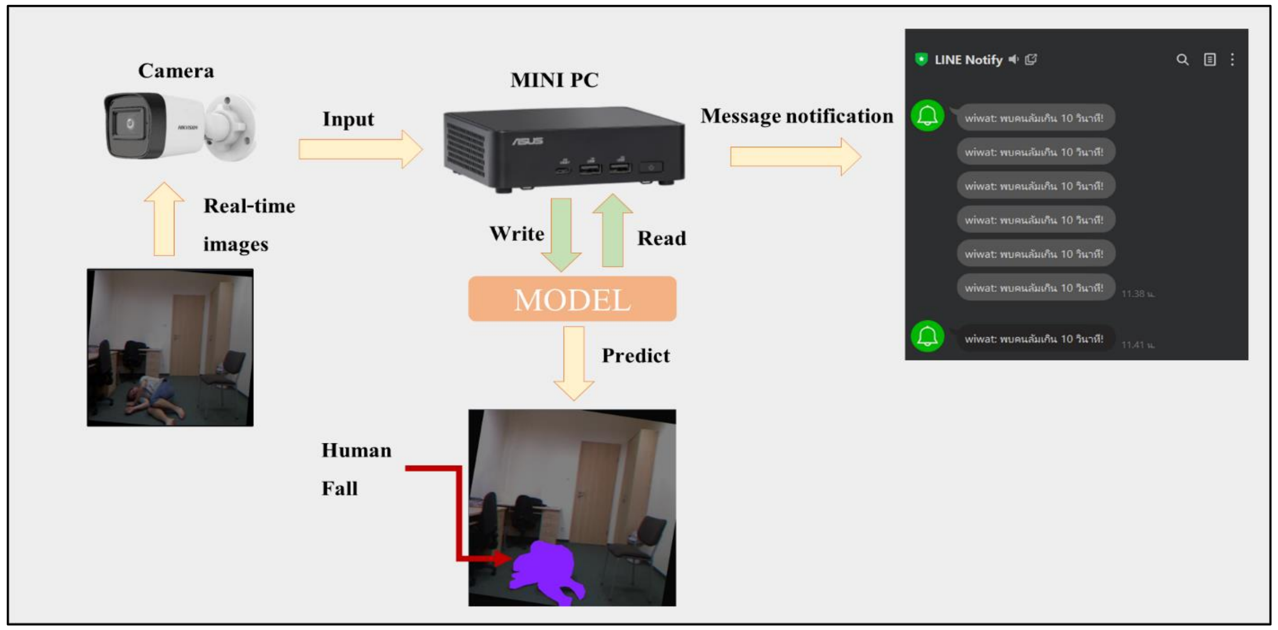

Figure 3. Principle of operation.

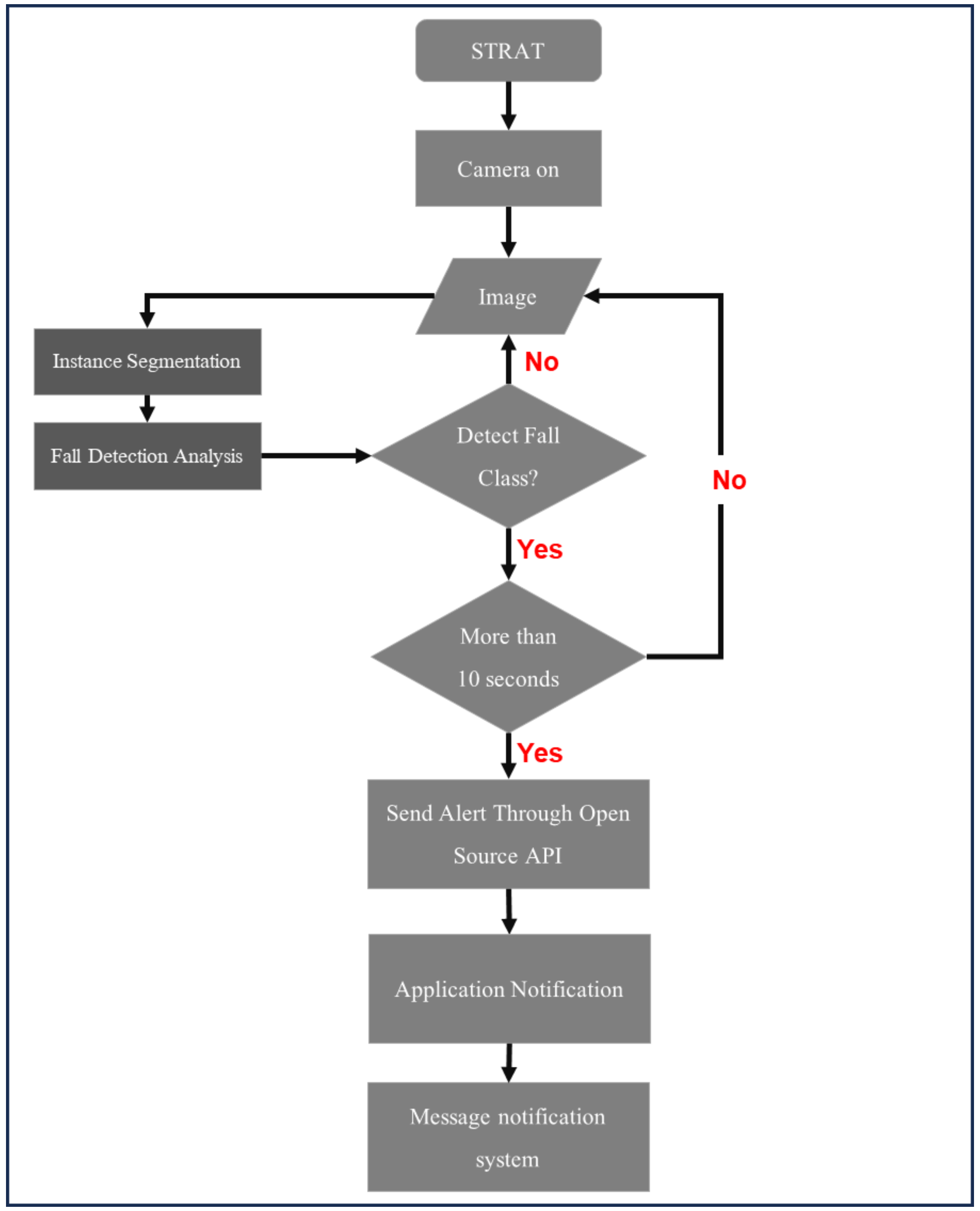

Figure 4. Flowchart.

4. Results and Discussion

In this paper, we propose the detection of fallen people by an instance segmentation method using the YOLOv11-seg model, which compares three models, namely YOLOv11s-seg, YOLOv11n-seg, and YOLOv11m-seg, by using Precision (P), Recall (R), mPA@0.5, and F1-scores to evaluate models and choose the right model for the research. The Precision (P) values of each model are 0.996, 0.929, and 0.934. The Recall (R) values are 0.850, 0.847, and 0.836, and the mPA0.5 values are 0.906, 0.906, and 0.763, respectively, as shown in Table 5. The results show that the suitable model is YOLOv11s-seg, as it achieves higher Precision, Recall, and mPA@0.5 than other models. In addition, there are fewer internal parameters than in other models. This makes the processing faster.

Table 5. Model Evaluation Results

Model	P	R	mAP@0.5
Yolov11s	0.996	0.850	0.906
Yolov11n	0.929	0.847	0.906
Yolov11m	0.934	0.836	0.763

To further analyze model behavior, Precision–confidence, Recall–confidence, and F1-score curves were examined, as shown in Figures 5–7. Across all three models, the Precision curves increase with higher confidence thresholds, while the F1-score curves decrease. This behavior reflects the expected trade-off between detection confidence and recall and indicates consistent and stable prediction characteristics across different model variants.

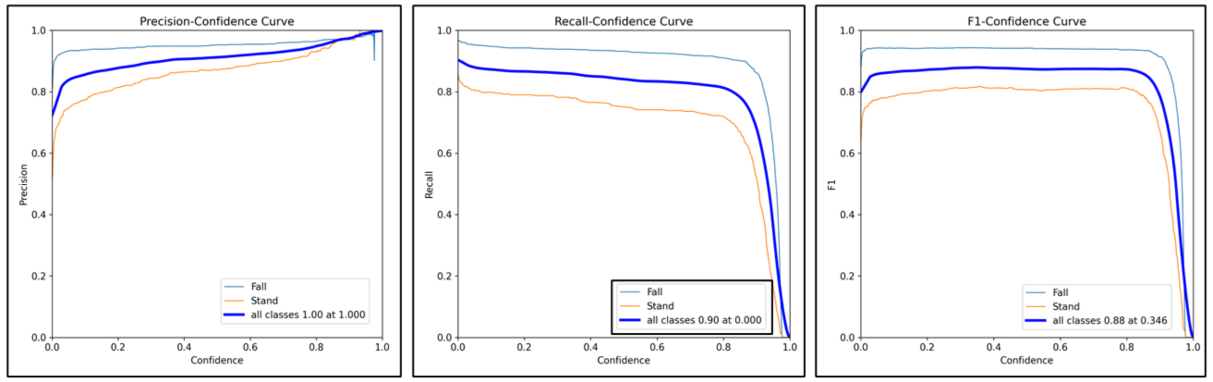

Figure 5. P-curve, R-curve, and F1-Scores curve of the YOLOv11s-seg model.

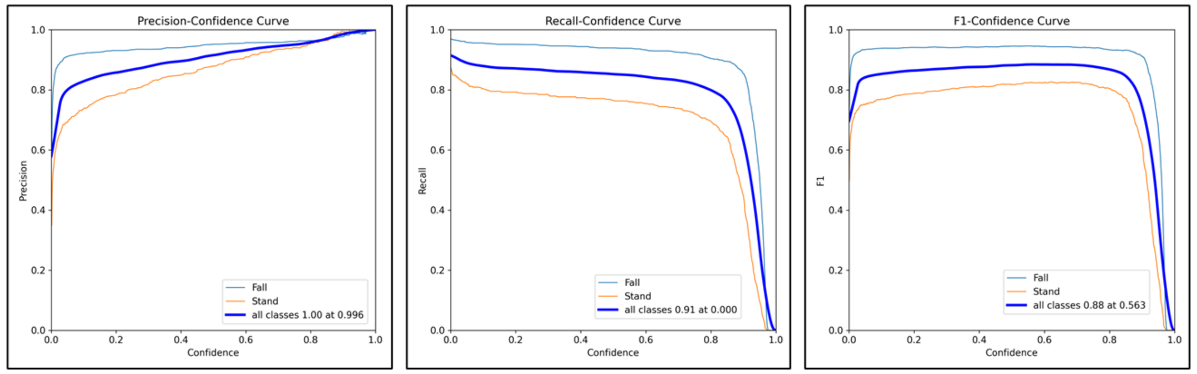

Figure 6. P-curve, R-curve, and F1-Scores curve of the YOLOv11n-seg model.

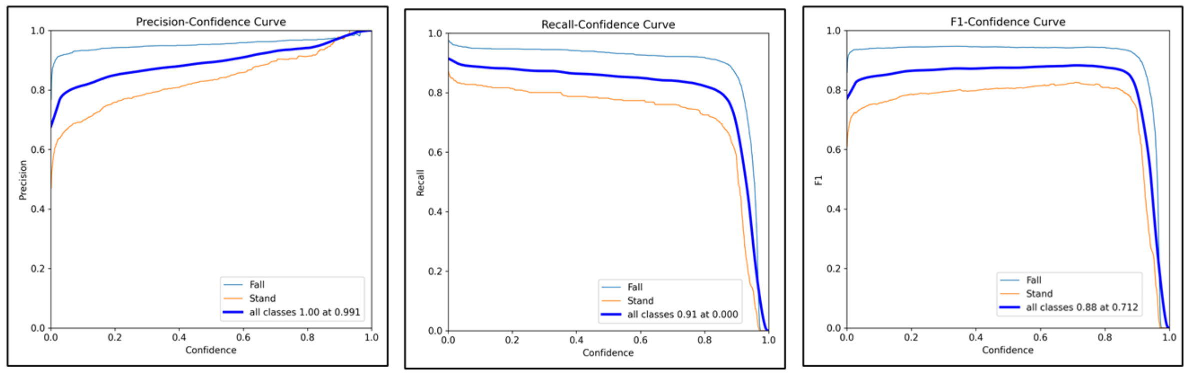

Figure 7. P-curve, R-curve, and F1-Scores curve of the YOLOv11m-seg model.

Qualitative evaluation results demonstrate that the proposed system can reliably distinguish between standing and falling postures under various environmental conditions. The system remains effective in challenging scenarios, including overlapping individuals and low-light environments. Representative detection results are illustrated in Figures 8 and 9, which show standing posture detection under normal and low-light conditions, respectively. Figures 10 and 11 present examples of fall detection for different fall orientations, including backward and inverted falls, under both lighting conditions. In all cases, the system visualizes the detected posture using segmentation masks and displays the corresponding confidence score, allowing users to assess detection reliability.

To position the proposed system within the broader context of fall detection research, Table 6 compares it with representative categories of existing fall detection approaches. Traditional computer vision–based methods offer low computational cost but suffer from limited robustness under illumination changes and occlusion. Wearable sensor–based systems achieve high detection accuracy but require continuous user compliance and device usage. Pose estimation–based and temporal deep learning approaches improve posture understanding and motion modeling, but introduce sensitivity to occlusion or increased computational complexity and latency. In contrast, the proposed YOLOv11-seg–based instance segmentation approach achieves high detection performance with real-time inference by directly analyzing pixel-level posture information, making it well-suited for edge and IoT-based fall detection applications.

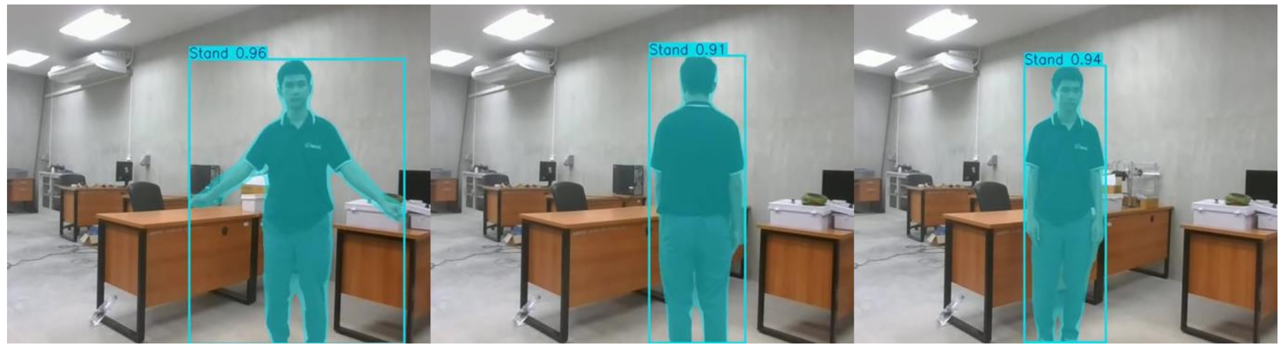

Figure 8. Human standing Detection in normal environmental conditions.

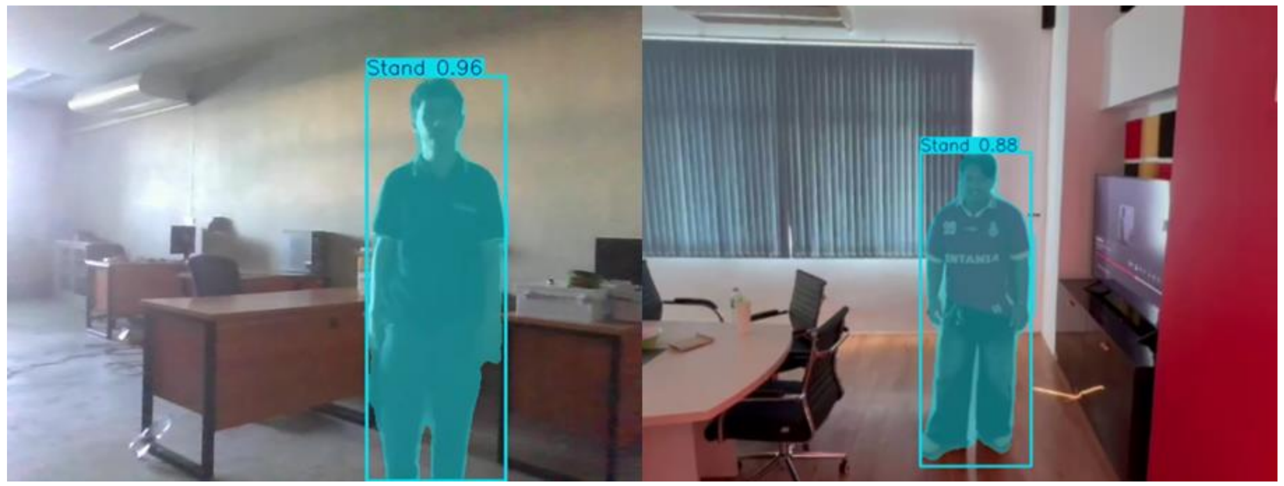

Figure 9. Human standing detection in low-light environmental conditions.

Figure 10. Detecting falls in various postures in normal environmental conditions.

Figure 11. Detecting falls across various postures in low-light environments.

In terms of performance reliability, the experimental results show that the model produces consistent outcomes across the training, validation, and testing datasets, indicating stable behavior when applied to unseen data. An analysis of detection errors reveals that false positive cases mainly occur during transitional human motions, such as bending or sitting, which may temporarily resemble falling postures. In contrast, false negatives are primarily observed in situations with partial occlusion or uncommon camera viewpoints, where the full body posture cannot be clearly captured.

Table 6. Comparison with Existing Fall Detection Approaches.

Approach Category	Representative Methods	Key Characteristics	Limitations	Suitability for Real-Time Deployment
Traditional Computer Vision [23]	Background subtraction, Optical flow, Motion history images	Low computational cost; simple implementation	Highly sensitive to illumination changes, occlusion, and background dynamics; limited robustness	Limited
Wearable Sensor-Based [24, 25]	Accelerometer, Gyroscope-based systems	Direct motion sensing; independent of visual conditions	Requires user compliance, discomfort, battery constraints, and privacy concerns	Moderate
Pose Estimation-Based [26]	OpenPose-based, skeleton tracking methods	Explicit posture and joint analysis; interpretable features	Degraded performance under occlusion, low resolution, or inaccurate key-point detection	Moderate
Temporal Deep Learning Models [27]	CNN–LSTM, TCN-based approaches	Captures motion dynamics over time; improved temporal consistency	Requires sequential buffering, higher computational overhead, and increased latency	Limited to Moderate
Proposed Method	YOLOv11-seg (Instance Segmentation)	Pixel-level posture representation; frame-level decision; robust to occlusion and illumination variation	Limited dataset diversity; absence of temporal modeling	High

5. Conclusions

This study presented a real-time human fall detection system based on instance segmentation using the YOLOv11-seg model. By analyzing human posture at the pixel level, the proposed system can distinguish between standing and falling postures under various environmental conditions. The integration of edge-based processing with an IoT notification mechanism enables timely alert delivery after a fall event is confirmed. A comparative evaluation of YOLOv11s-seg, YOLOv11n-seg, and YOLOv11m-seg showed that YOLOv11s-seg offers the best balance between detection performance and computational efficiency. The model achieved higher Precision, Recall, and mAP@0.5 values while maintaining a lower parameter count, making it appropriate for real-time deployment on edge devices. Several limitations should be acknowledged. The dataset was collected mainly from a university environment and includes only two posture classes, which may limit generalizability. In addition, the current system lacks temporal modeling, and formal statistical validation was not conducted. Future work will focus on expanding the dataset, incorporating additional posture classes and temporal information, and further evaluating the system in more diverse real-world environments.

6. Acknowledgements

We are grateful to the Faculty of Engineering, Thaksin University, Phatthalung Campus, for allowing the location for this research experiment.

Author Contributions: The following are the contributions of each author in this research: Conceptualization: Tanawat Srirugsa and Saowanee Singsarothai; Methodology: Tanawat Srirugsa and Tawat Chuchit; Software: Wiwat Su-hren and Supachai Kaewpoung; Writing—review and editing: Tawat Chuchit and Wiwat Su-hren; Project administration: Tawat Chuchit.

Funding: This work was supported by the Fundamental Fund for fiscal year 2025, provided by the Science, Research and Innovation Promotion Fund, for the project "Design and Development of a Smart Vegetable Plot Prototype with Internet of Things (IoT) Technology for the Elderly" at Thaksin University.

Conflicts of Interest: The authors of this study hereby state that they have no conflicts of interest.

References

[1] World Health Organization. Falls. https://www.who.int/news-room/fact-sheets/detail/falls (accessed 2026-02-23).

[2] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *Proc. IEEE Int. Conf. Comput. Vis.* **2017**, 2980–2988. https://doi.org/10.1109/ICCV.2017.322

[3] Zhang, S.-H.; Li, R.; Dong, X.; Rosin, P. L.; Cai, Z.; Han, X.; Yang, D.; Huang, H.-Z.; Hu, S.-M. Pose2Seg: Detection Free Human Instance Segmentation. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* **2019**, 8811–8819. https://doi.org/10.1109/CVPR.2019.00902

[4] Lu, N.; Wu, Y.; Feng, L.; Song, J. Deep Learning for Fall Detection: Three-Dimensional CNN Combined With LSTM on Video Kinematic Data. *IEEE J. Biomed. Health Inf.* **2019**, *23*(1), 314–323. https://doi.org/10.1109/JBHI.2018.2808281

[5] Yao, L.; Yang, W.; Huang, W. A Fall Detection Method Based on a Joint Motion Map Using Double Convolutional Neural Networks. *Multimed. Tools Appl.* **2022**, *81*, 4551–4568. https://doi.org/10.1007/s11042-020-09181-1

[6] Su, S.; Wu, S.-S.; Chen, S.-Y.; Duh, D.-J. Multi-view Fall Detection Based on Spatio-temporal Interest Points. *Multimed. Tools Appl.* **2016**, *75*, 8469–8492. https://doi.org/10.1007/s11042-015-2766-3

[7] Gao, M.; Li, J.; Zhou, D.; Zhi, Y.; Zhang, M.; Li, B. Fall Detection Based on OpenPose and MobileNetV2 Network. *IET Image Process.* **2023**, *17*(3), 722–732. https://doi.org/10.1049/ipr2.12667

[8] Alam, E.; Sufian, A.; Dutta, P.; Leo, M. Vision-Based Human Fall Detection Systems Using Deep Learning: A Review. *Comput. Biol. Med.* **2022**, *146*, 105626. https://doi.org/10.1016/j.compbiomed.2022.105626

[9] Li, L.; Fang, M.; Fu, F. Instance Segmentation Based on Improved YOLACT. *Proc. Int. Conf. Virtual Real. Vis. (ICVRV)* **2020**, 165–170. https://doi.org/10.1109/ICVRV51359.2020.00042

[10] Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725. https://doi.org/10.48550/arXiv.2410.17725

[11] He, L.; Zhou, Y.; Liu, L.; Ma, J. Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites. *Buildings* **2024**, *14*(12), 3777. https://doi.org/10.3390/buildings14123777

[12] Deepa, D.; Sivasangari, A.; Roonwal, R.; Nayan, R. Pothole Detection using Roboflow Convolutional Neural Networks. *Proc. 7th Int. Conf. Intell. Comput. Control Syst. (ICICCS)* **2023**, 560–564. https://doi.org/10.1109/ICICCS56967.2023.10142700

[13] Su-Hren, W.; Chuchit, T.; Kaewpoung, S. Development of an Autonomous EV Navigation System Using Instance Segmentation for Medical Supply Delivery. *Proc. 13th Int. Electr. Eng. Congr. (iEECON)* **2025**, 1–6. https://doi.org/10.1109/iEECON64081.2025.10987580

[14] Babila, I. F. E.; Villasor, S. A. E.; Dela Cruz, J. C. Object Detection for Inventory Stock Counting Using YOLOv5. *Proc. IEEE 18th Int. Colloq. Signal Process. Appl. (CSPA)* **2022**, 304–309. https://doi.org/10.1109/CSPA55076.2022.9782028

[15] Mostafa, M.; Sadi, S.; Anamika, S. A.; Hussain, M. S.; Khan, R. Automatic Vehicle Classification and Speed Tracking. *Proc. 2nd Int. Conf. Appl. Artif. Intell. Comput. (ICAAIC)* **2023**, 972–977. https://doi.org/10.1109/ICAAIC56838.2023.10140935

[16] Pornpanomchai, C. Image Based Papaya (*Carica papaya* Linn.) Seed Germination Evaluation by ResNet50. *ASEAN J. Sci. Technol. Rep.* **2025**, *28* (1), e254804. https://doi.org/10.55164/ajstr.v25i1.245292

[17] Elaanba, A.; Ridouani, M.; Hassouni, L. Automatic Detection Using Deep Convolutional Neural Networks for Abnormal Positioning of Tubes and Catheters in Chest X-ray Images. *Proc. IEEE World AI IoT Congr. (AIIoT)* **2021**, 7–12. https://doi.org/10.1109/AIIoT52608.2021.9454205

[18] Ahmed, S.; Rahman, M. J.; Razzak, M. A. Design and Development of an IoT-Based LPG Gas Leakage Detector for Households and Industries. *Proc. IEEE World AI IoT Congr. (AIIoT)* **2023**, 762–767. https://doi.org/10.1109/AIIoT58121.2023.10174377

[19] Karar, M. E.; Shehata, H. I.; Reyad, O. A Survey of IoT-Based Fall Detection for Aiding Elderly Care: Sensors, Methods, Challenges and Future Trends. *Appl. Sci.* **2022**, *12*(7), 3276. https://doi.org/10.3390/app12073276

[20] Mrozek, D.; Koczur, A.; Małysiak-Mrozek, B. Fall Detection in Older Adults with Mobile IoT Devices and Machine Learning in the Cloud and on the Edge. *Inf. Sci.* **2020**, *537*, 132–147. https://doi.org/10.1016/j.ins.2020.05.070

[21] Yacchirema, D.; de Puga, J. S.; Palau, C.; Esteve, M. Fall Detection System for Elderly People Using IoT and Ensemble Machine Learning Algorithm. *Pers. Ubiquitous Comput.* **2019**, *23*, 801–817. https://doi.org/10.1007/s00779-018-01196-8

[22] Stanivuk, I.; Bjelić, V.; Samardžić, T.; Simić, Đ. Expanding Lua Interface to Support HTTP/HTTPS Protocol. *Proc. 13th Int. Conf. Adv. Technol., Syst. Serv. Telecommun. (TELSIKS)* **2017**, 407–410. https://doi.org/10.1109/TELSKS.2017.8246311

[23] Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Fall Detection from Human Shape and Motion History Using Video Surveillance. *Proc. 21st Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINAW)* **2007**, 875–880. https://doi.org/10.1109/AINAW.2007.181

[24] Mubashir, M.; Shao, L.; Seed, L. A Survey on Fall Detection: Principles and Approaches. *Neurocomputing* **2013**, *100*, 144–152. https://doi.org/10.1016/j.neucom.2011.09.037

[25] Noury, N.; Fleury, A.; Rumeau, P.; Bourke, A. K.; Laighin, G. O.; Rialle, V.; Lundy, J. E. Fall Detection – Principles and Methods. *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS)* **2007**, 1663–1666. https://doi.org/10.1109/IEMBS.2007.4352627

[26] Cao, Z.; Simon, T.; Wei, S.-E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* **2017**, 1302–1310. https://doi.org/10.1109/CVPR.2017.143

[27] Yang, G.; Zhao, J.; Guo, J. Research on Fall Detection Algorithm Based on CNN and LSTM. *Proc. 3rd Int. Conf. Nat. Lang. Process. (ICNLP)* **2021**, 190–195.