EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 6 No. 2 (2025) 70-82
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Development of a Child Fall Detection System based on YOLOv9 using Transfer Learning from Adult Fall Datasets

Nurul Khalidah Mohd Yusof[1], Suhaila Sari[1*], Nik Shahidah Afifi Md Taujuddin[1], Hazli Roslan[2], Nabilah Ibrahim[1]

[1] *Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA*

[2] *Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia, Pagoh, 84600, MALAYSIA*

*Corresponding Author: suhailas@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2025.06.02.009

Article Info

Received: 26 June 2025
Accepted: 04 August 2025
Available online: 30 October 2025

Keywords

YOLOv9, fall detection, deep learning, transfer learning, child safety.

Abstract

Falls detection systems are vital for ensuring safety, particularly among vulnerable groups such as children. However, most existing systems are trained using adult-specific datasets, limiting their effectiveness for child fall scenarios due to differences in body size, movement patterns, and fall dynamics. In this study, a child fall detection system was developed using YOLOv9, a state-of-the-art object detection algorithm, with transfer learning applied from adult fall datasets. The system was built through several stages including dataset collection, annotation using Roboflow, model training in Google Colab, and deployment via a user interface created using Gradio. Three experimental setups were conducted: training and testing on adult datasets, testing on child fall images using a model trained solely on adult data, and testing on multi-subject child scenarios. The YOLOv9 model was trained and evaluated at multiple epochs (30, 50, 70, and 100) using standard performance metrics such as precision, recall, F1 score, and mAP. Results showed that while the model achieved high accuracy on adult data (mAP@0.5 up to 0.995), it also demonstrated reasonable performance in detecting child falls, especially at 100 epochs, with a precision of 0.98 and recall of 1.00 in single-child scenarios. Although performance declined slightly in multi-subject scenes, the findings confirm that the developed system can generalize from adult to child fall detection with acceptable accuracy. This study demonstrates the possibility of using transfer learning in child safety as well as providing a cost-effective solution in those settings where a child-specific data is scarce.

1. Introduction

Transfer Learning YOLOv9 the system proposed in this project to develop a child fall detection system based on a transfer learning idea with results of fall detection in adults will help to fill this gap and resolve this important issue of safety monitoring in real-time. Accidental falls in children are a major cause of injury, it is therefore necessary to identify it early so that interventions are administered as soon as possible to reduce damages. The conventional fall detection mechanisms have usually used wearable sensors that are annoying and produce false alarms especially in active children. Consequently, non-invasive techniques based on deep learning and object detection have improved in the field of vision due to their high precision [1].

The most recent project in the You Only Look Once (YOLO) family, YOLOv9, proposes an architectural innovation in the form of a Generalized Efficient Layer Aggregation Network (GELAN) and Programmable Gradient Information (PGI), the former and the latter technique improving feature distillation and network flow gradients, respectively, improving the accuracy and computational cost of detecting objects on their original versions [2]. Such advances render YOLOv9 especially applicable to real-time usage, such as at the edge, on resource-conscious edge devices, which is essential in the application of a child monitoring system in practice [2] [3].

One of the most meaningful problems on the way to creating child-specific fall detection systems is the lack of annotated datasets including children. To get past this, the transfer-learning technique is used and in transfer-learning a model named YOLOv9 which is pre-trained on large adult fall datasets is fine-tuned on a limited available data of child falls. Such an approach could be called as utilizing previously known about human poses and dynamics of falling, having the model follow their patterns and adapt to some new patterns and poses of children with the minimal additional information. Experiments conducted have established that these methods of transfer learning can enhance the generalization and detecting abilities of the model significantly even with occlusions as well as differing lighting and intricate backgrounds [4] [5].

The latest studies demonstrate that by incorporating multi-scale feature extraction, attention mechanism, and cost-effective losses work, the system also improved the capability to recognize falls occurring in even the problematic settings locating and registering them promptly [4]. The outcome is a strong, lightweight, and scaleable fall detection solution that will not only enhance the child safety but also support real time notifications and interventions and as such is well suited to application in homes, schools, and childcares [1] [3] [4].

2. Literature Review

Falls are a major issue to many groups of people especially the old and the small children because of the high risk of sustaining major injury and long time health effects. As a reaction to this, fall detection systems (FDS) have turned out to be a major topic of investigation and have developed substantially within the past decades. First, FDS was based on the threshold method of wearable sensors including accelerometers and gyroscopes in identifying sudden changes in movement. Yet, these early methods had drawbacks of too many false positives and too little real-life generalizability since most data sets were collected on young adults falling in simulated conditions. To avoid these shortcomings, investigators have resourcefully resorted to the computer vision and artificial intelligence (AI), exploiting advanced algorithms and deep learning. The vision based solutions especially the one using convolutional neural networks (CNNs) and object detectors based on YOLO make accurate distinction between falls and other normal movements with potential outcomes of extracting complex features utilizing automated video data sceneries. Nevertheless, there are obstacles to overcome, such as a lack of a variety and real-world data, in particular, data on children; issues related to privacy, efficiency, and the usage of this type of learning in real-time. Literature identifies an increase in the use of multi-modal, machine learning-based FDS which incorporates technologies in Internet of Things (IoT) and edge computing to allow monitoring without interruption and without intrusion. The report does an overview of key studies in the field, highlighting the methodologies, the strengths and weaknesses of the current strategies and the significance of coming up with strong and generalizable systems of multiple environments and age group [6][7][8].

2.1 Flexible Fall Detection Framework Based on Object Detection and Motion Analysis

Such a framework combines YOLOv5 to detect objects and high-level motion analysis, which does not require any additional tools. manual tuning of thresholds. It is able to recognize falls by following persons in several video frames. fluctuations in the silhouette ratio, orientation variations and centroid fluctuations. Markedly, the system shows better flexibility and copes with different camera arrangements and angles. As an example, there are assessments of its soundness ascertained by such datasets as Le2i and URFD; the datasets are more precise and scalable in comparison to the traditional ones. methods. Therefore, this makes the MMEFD framework a viable solution to fall detection in dynamic surveillance environments [9]. The extracts of test video datasets are depicted in Fig. 1 by the sample snapshots.

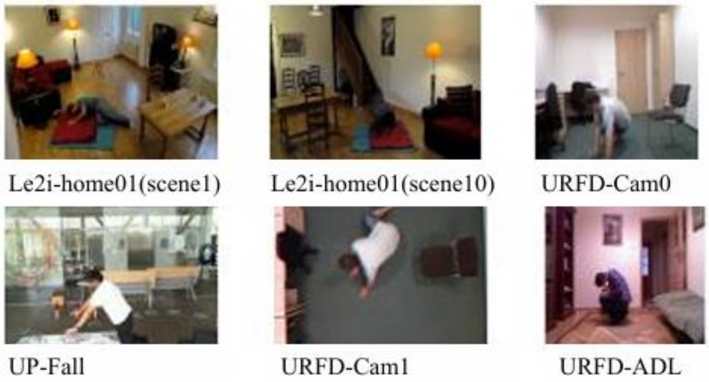

Fig. 1 *Sample snapshots of test video datasets*[*9*]

2.2 Human Fall Detection Algorithm Based on YOLOv3

This project will also focus on developing and simulating the use of a real-time fall-sensing mechanism that is precise and effective, and whose end-target will be to better the safety-check among at-risk people specifically the older population [10]. The project is based on the use of the YOLOv3 deep learning algorithm that will allow fast detection of human falls in any complex situation. It is trained, checked and tested on an adult picture database [10]. To do so, the authors created a personalized dataset of 2,600 images labeled in a Pascal VOC format and built especially with fall detection tasks in mind. YOLOv3 model whose foundation is the Darknet-53 architecture was made to be optimized on the basis of the K-means cluster method to coordinate the refinement of anchor boxes to improve the localization of objects. The GPU environment provided an effective model that was able to achieve a high value of mean average precision (mAP) of 0.83 and a fall detection accuracy of 97%. These findings indicate the usefulness of the suggested system in the recognition of falls in different conditions. As it was shown in Fig. 2, the system correctly determines a case of "Not fall" [10], whereas Fig. 3 gives an example of a successful detection of a case of "Fall" [10]. The fact that this paper shows the possibility of incorporating YOLOv3 into health monitoring systems so that to guarantee a response time in emergencies considerably quicker which in its turn may lead to better care and safety of endangered people.

Fig. 2 *"Not fall" test result* [*10*]

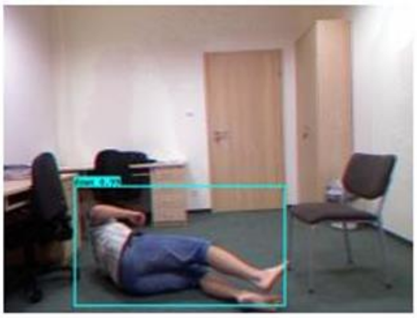

Fig. 3 *"Fall" test result* [*10*]

2.3 An image-based Fall Detection System for the Elderly Using YOLOv5

The system is combined to create a non-intrusive and reliable real-time fall detection system that can positively change the way elderly people enjoy their lives, particularly those living alone [11]. This project is aimed at developing efficient use of advanced image processing and deep learning models (YOLOv5 object detection model) to detect human falls based on visual analysis furthermore, it does not rely on the use of wearable technologies or special sensors. It also utilizes an adult image dataset for training, validation, and testing [11]. The key point of the project is to create a system that accurately distinguishes between different human postures (such as standing, sitting, and falling) using labeled image data, thereby minimizing false alarms and improving response time in emergency situations. As shown in Fig. 4, the results obtained demonstrate the model's capability to correctly identify various human postures and fall scenarios across different images [11]. By eliminating the need for continuous human monitoring or uncomfortable sensor-based systems, the project aims to offer a scalable, efficient, and cost-effective solution to address one of the leading causes of injury and death among the elderly population.

Fig. 4 *Obtained results on various images* [*11*]

3. Methodology

In Fig. 5, the process illustrates the block diagram of a Fall Detection System using YOLOv9 object detection architecture. The system begins with the input phase, where a total of 120 adult and child fall images are collected and annotated for model training. These images are obtained from open-source and self-collected datasets, which are labeled using Roboflow, a popular annotation tool. Preprocessing steps such as resizing (to 440×440 pixels), brightness adjustment, and zoom augmentation are applied to standardize the data and enhance model generalization. This study performed three experiments to evaluate the effectiveness of a YOLOv9-based fall detection system trained on adult fall images and its ability to generalize to child fall scenarios. The experiments utilize a datasets division of 80% for training, 10% for validation, and 10% for testing.

The control component of the system is managed by the YOLOv9c model, which is trained on the adult fall image dataset. The training is performed on Google Colab, with configurations including multiple epochs (30, 50, 70, and 100) to observe performance trends. The model is responsible for detecting fall events from images or video frames, including both adults and children, even though it is trained solely on adult data. This approach evaluates the feasibility of cross-domain learning using transfer learning principles.

The detection process outputs bounding boxes around fall events along with confidence scores. If a fall is detected, the system transitions to the output phase, where the fall detection status is recorded and displayed.

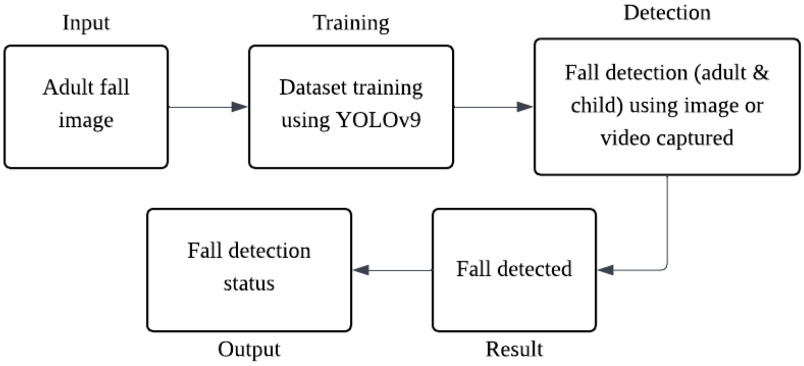

Fig. 5 *Block diagram of child fall detection using YOLOv9 trained on the adult fall dataset*

Fig. 6 illustrates the complete workflow of the fall detection system development process. The adult and children fall image dataset is collected (Kaggle and self-collected). The images are then annotated using Roboflow to identify key features relevant to fall detection.

The annotated dataset is prepared and divided into three subsets: 80% for training, 10% for validation, and 10% for testing. Such subsets are applied in different phases of developing models. Object detection models An object detection model selects the YOLOv9. The quantity of epochs and the batch size can be regarded as hyper-parameters which must be set before the training process.

The model is then made to learn with training set and the performance is watched. The model is tested in terms of effectiveness after it is trained by comparing it to the validation and test datasets. The model performance is evaluated on the metrics of its precision, recall, F1-score, and mAP@0.5. The outcomes aid in establishing the credibility of the model of detecting falls in children without involving training data specific to children. Lastly the system gives the fall detection status that is, tells whether fall detection has occurred successfully hence terminating the process.

3.1 Annotation of the Dataset

An important element of the training process of developing the model was the need to annotate fall images so as to provide the right and contextually relevant training data. Exact labels are needed to be able to train the model to discriminative visual cues which go hand in hand with fall cases. In the current work, the adult fall records were annotated using Roboflow.ai platform.

The set-up annotation process started by opening a project workspace on Roboflow and uploading the appropriate images on representational fall situations. The manual annotation of the images was done by defining the bounding boxes and labeling them with the class name = fall. This categorization assists the model to recognize and distinguish fall patterns on training. With regular labeling of fall-specific topics, the system can tie particular visual information to the occurrence of falls and this increases chances of generalizing to new images hence.

Fig. 7 shows the example of the annotated photo in Roboflow.ai in which the fall subject is labeled with a bounding box to be used in the training process.

3.2 Graphical User Interface Using Gradio

To demonstrate the results of the fall detection in a format easy to understand, this work used Gradio, which is an open-source library written in Python and aimed at making the interaction with machine learning models possible using a web interface. Gradio is an easy-to-use graphical user interface (GUI) that allows the real-time connection to the trained YOLOv9 model, with the possibility of uploading or choosing an image and seeing the output result with confidence scores in real-time. This removes the step of web development since Gradio does the frontend development with as little Python code as possible. It accepts all kinds of input and output data (i.e., text, images, sliders, etc.) and is easily combined with other machine learning ecosystems (i.e., TensorFlow, PyTorch, scikit-learn, etc.). The interface, Gradio, used in this project is intended to make it easy to be deployed and tested quickly with the possibility of creating public and sharable links to directly access environments, such as Google Colab. As delivered in Fig. 8, the fall detecting system with the gradio protocol is trivial and user-friendly, easy to test and use even by the individuals without technical skills. Simplicity, flexibility, and capability of giving real-time feedback, makes Gradio a powerful instrument in implementing AI models into interacting applications [12][13][14][15].

3.3 Training, Validation and Testing of YOLOv9

The process of training, validating and testing the YOLOv9 model in detecting the children falls based on an adult fall-detection dataset can be addressed as the set of ordered steps performed within Google Colab. Start the procedure with the opening of Google Colab and the creation of a new notebook. To train the models faster and to enhance the computational speed, one must provide GPU support by following the steps as below: Click the dropdown on the toolbar on 'Run as' and select 'Change runtime type' option and change to T4 GPU as the hardware accelerator.

The second stage is pre-processing the data with regards to accessing annotated images and labels. This is achieved through the mounting of Google Drive to the Colab environment so that the necessary training, validation, and testing files, as well as the YAML configuration, can be accessed without hassles. One can upload these datasets either manually or can access it directly through the integration of Roboflow where the pre-labeled dataset is saved in the form of YOLOv9.

After the dataset was made publicly and conveniently available via Roboflow and after it was preprocessed adequately, it was split into the training, validation, and testing groups so that a thorough analysis of its possible performance can be conducted. In this project we used the YOLOv9 object detection architecture that did not need the manual extraction of features as opposed to the classic CNN models. YOLOv9, in turn, learns the structure of spatial hierarchies in the image frames, so the detections are faster and more accurate. Having trained the YOLOv9 on the adult fall images, the second step was carrying out the evaluation test that transcended testing the model in the adult and offspring contexts in fall detection. A Python code was used to load test images using Google Colab, process the test images through a trained YOLOv9 model, then confirm if the label fall occurred in the model predictions. Names of the classes were received and compared and in case of a fall, a result was logged in the system in that case.

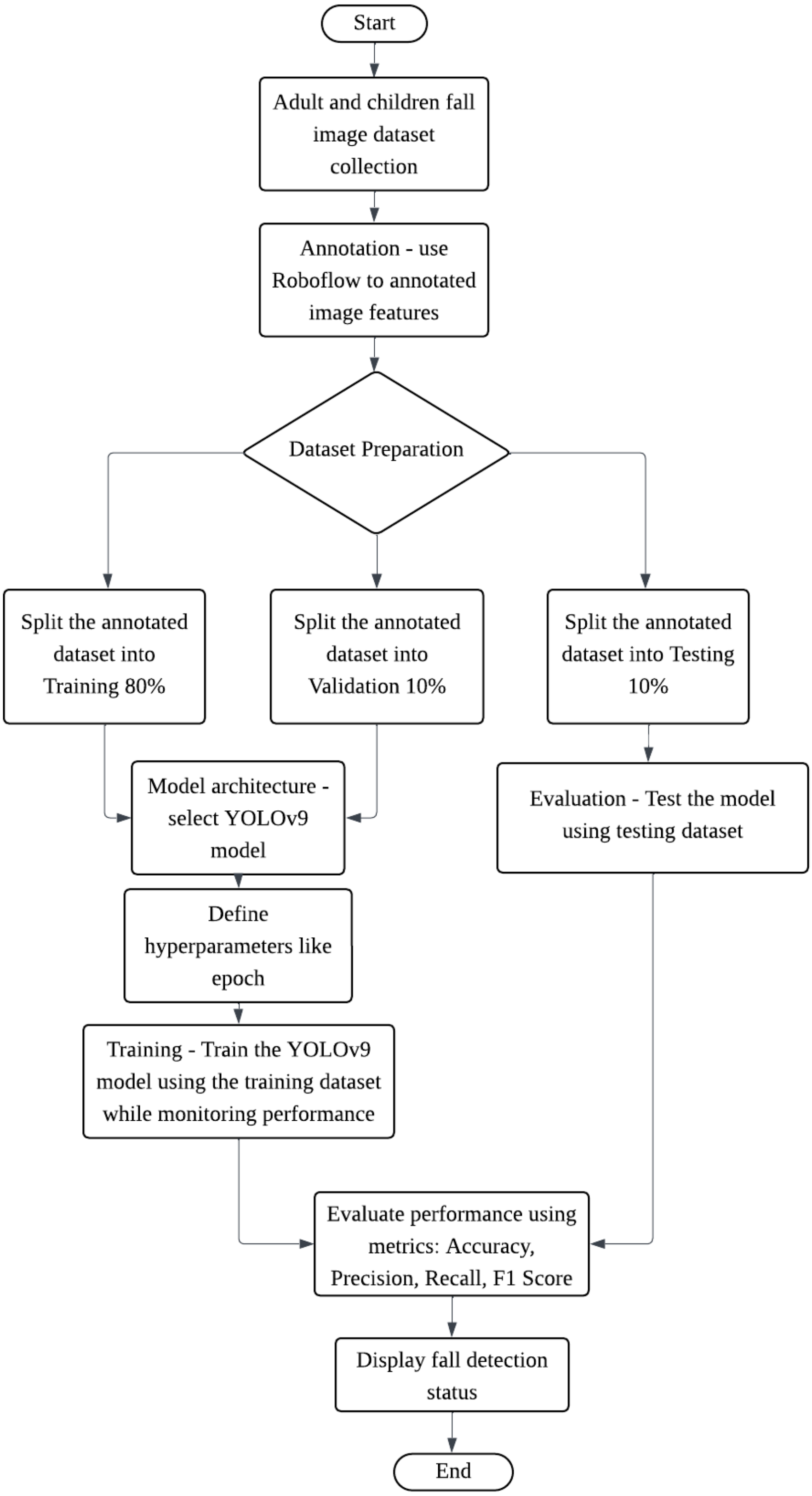

Fig. 6 *Flowchart of the project*

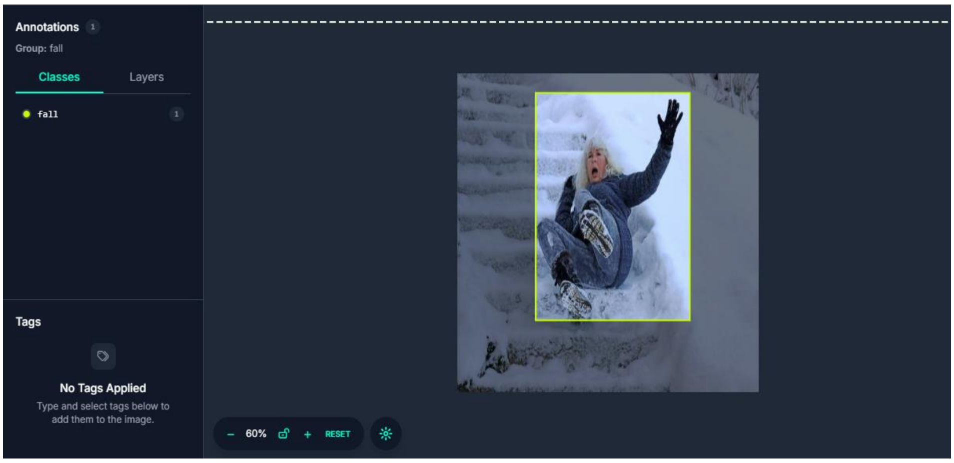

Fig. 7 *Label in Roboflow.ai*

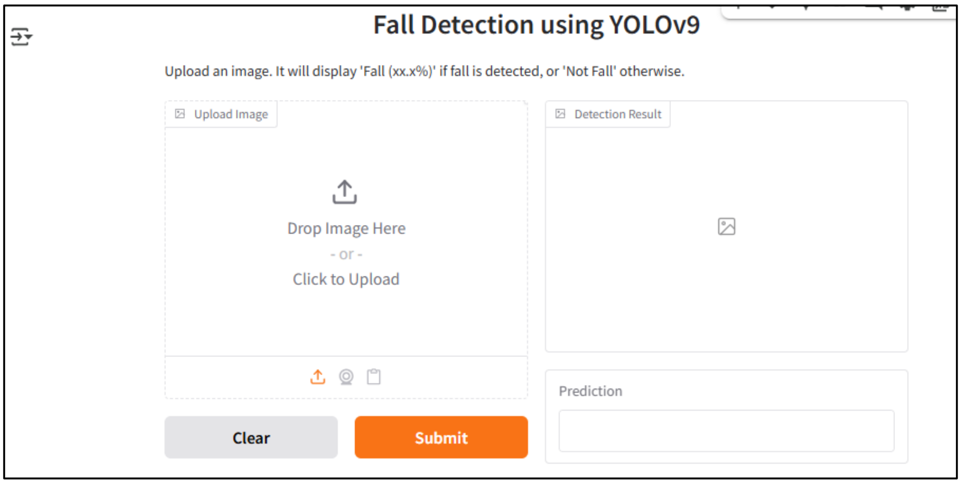

Fig. 8 *Main interface of the Gradio-based fall detection system*

4. Results and Discussion

This study conducted three experiments to evaluate the effectiveness of a YOLOv9-based fall detection model trained on adult images and its ability to generalize to child fall scenarios. In Experiment 1, the model was trained and tested using 100 adult fall images to set a baseline (Training (80%) – 80 adult fall images, Validation (10%) – 10 adult fall images, and Testing (10%) – 10 adult fall images). Experiment 2 tested the same model on single-child fall images using 110 adult and child fall images dataset (Training (80%) – 88 adult fall images, Validation (10%) – 11 adult fall images, and Testing (10%) – 11 child fall images (each image includes a single child in a fall situation)). Experiment 3 tested it on more complex scenes involving multiple children using 120 adult and children fall images dataset (Training (80%) – 96 adult fall images, Validation (10%) – 12 adult fall images, and Testing (10%) – 12 children fall images (each image includes at least one child experiencing a fall in each image)). Dataset augmentation preprocessing techniques included resizing, normalization, and data augmentation (zoom and brightness adjustments) to improve generalization.

The YOLOv9 model was trained using four different epoch values (30, 50, 70, and 100), with a batch size of 16 and image resolution of 440×440 pixels.

Model performance was measured using accuracy, precision, recall, and F1-score. The results from Experiment 1 served as the benchmark to compare the model's ability to detect child falls in Experiments 2 and 3.

4.1 Investigation on the Performance of Adult Fall Detection Using the Adult Fall Dataset

In this part, the emphasis will be made on the performance analysis of the baseline version of information obtained using the Asus YOLOv9 algorithm trained and tested on the same adult fall dataset. There was a total of 100 images with description of adult fall scenarios, and it was divided into 80 percent training, 10 percent validation and 10 percent testing. The objective was to identify the ability of the model in detecting the fall incidences in a controlled and uniform area.

Four training epoch conditions (30, 50, 70, and 100) were used in training the model to note the dissimilarity in performance, across the training conditions. Each setup involved 16 round batches and an image of dimensions

440.440 pixels. To create generalization, the images were preprocessed and augmented, i.e., resized, zoomed, and adjusted with brightness. According to Figure 9, the YOLOv9 model can identify a situation of adult fall, so it demonstrates its possibilities to localize objects and to classify them inside the dataset.

Fig. 9 *Example of YOLOv9 detection result on adult fall image*

Fig. 10 presents the graphical user interface (GUI) output when an adult is present in the frame, but no fall is detected. This serves to validate the system's ability to correctly classify normal or non-fall activities, avoiding false positives and enhancing the model's reliability in real-world applications.

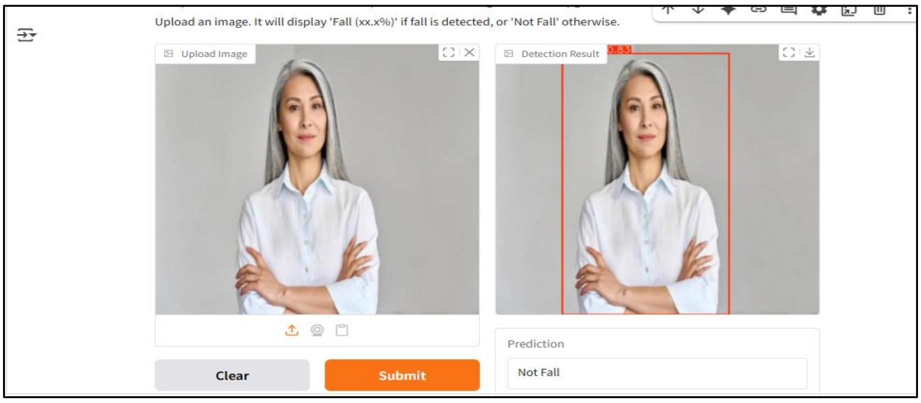

Fig. 10 *GUI output of an adult in non-fall condition. The system correctly identifies the scene with "Not Fall" detection triggered*

Fig. 11 shows the GUI output when an adult fall is detected. The bounding box highlights the fall event with an associated confidence score, demonstrating that the model can accurately detect and localize adult fall incidents when trained and tested within the same domain.

Fig. 11 *GUI output of adult fall detection. The fall is successfully identified and labeled with a confidence score*

Performance evaluation was based on standard object detection metrics: precision, recall, F1-score, and mean Average Precision (mAP@0.5). Table 1 shows the performance comparison of the YOLOv9 model at different training epochs on the adult fall dataset. The results indicated that the model achieved its highest performance at 50 epochs, with a precision of 0.984, a recall of 1.000, an F1-score of 0.992, and mAP@0.5 of 0.995. These metrics demonstrate that the YOLOv9 model is highly effective when applied to a dataset with consistent visual features, validating its reliability for adult fall detection in controlled environments.

Table 1 *Performance comparison of the YOLOv9 model at different training epochs on the adult fall dataset*

Metric	Epoch 30	Epoch 50	Epoch 70	Epoch 100
Precision	0.896	0.984	0.817	0.998
Recall	0.600	1.000	0.900	1.000
F1 Score	0.720	0.992	0.855	0.999
mAP@0.5	0.766	0.995	0.886	0.995
mAP@0.5:0.95	0.512	0.632	0.664	0.759

4.2 Investigation on the Performance of Child Fall Detection Using an Adult Fall Detection Training Dataset on a Child Test Dataset

This part examines the appropriate performance of the YOLOv9 model when identifying a situation of child falls trained on an adult fall dataset alone. This is aimed at determining the generalization capacity of the model using transfer learning method since gathering and annotation of actual data of child falls might be ethically and practically difficult to construct. For this experiment, a separate test dataset containing images of a single child involved in fall events was used.

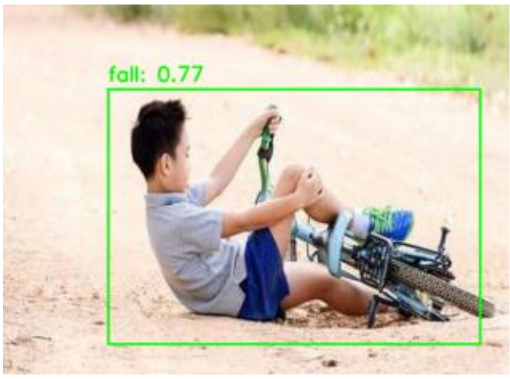

Fig. 12 *Sample result of child fall detection using a YOLOv9 model trained on adult data*

Fig. 13 displays the GUI interface when a child is shown performing non-fall activities. The absence of bounding boxes indicates that the model correctly identifies the scene as a non-fall situation, showing its ability to reduce false detections even when dealing with child images, despite being trained on adult data.

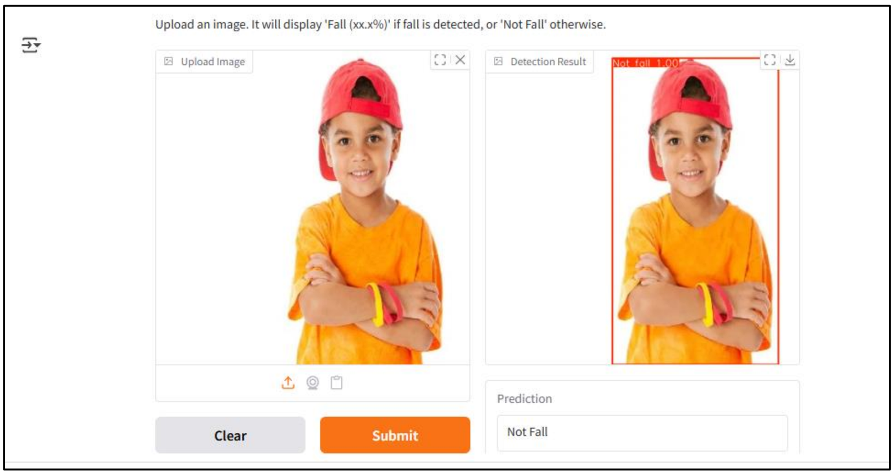

Fig. 13 *GUI output of single child in non-fall condition. The system accurately recognizes the absence of a fall*

Fig. 14 illustrates the GUI result for a child fall scenario involving a single subject. The bounding box drawn around the child signifies a successful detection by the model, confirming its generalization capability through transfer learning, even without direct training on child-specific fall data.

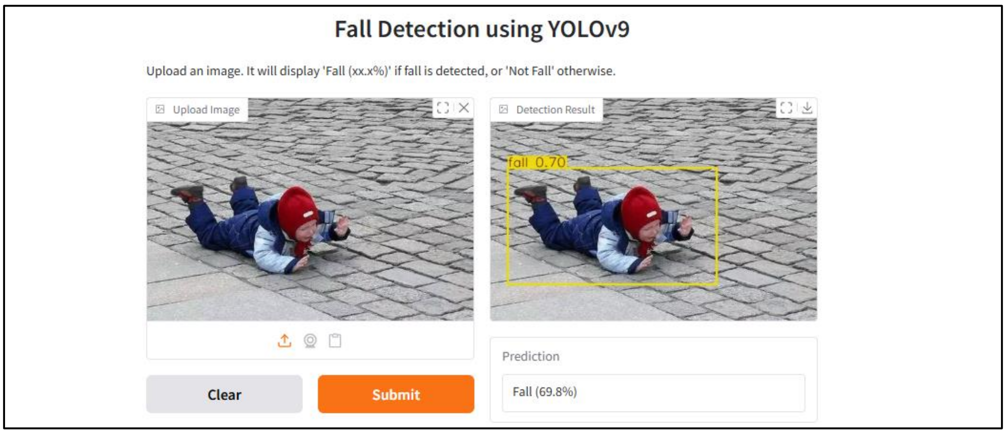

Fig. 14 *GUI output of single child fall detection. The fall is detected and annotated by the system*

The model was tested using the same configuration as in previous experiments, image resolution of 440×440 pixels, batch size of 16, and varying training epochs (30, 50, 70, and 100). Evaluation was conducted using key performance metrics, including precision, recall, F1-score, and mAP. Table 2 shows the performance comparison of the YOLOv9 model at different training epochs when tested on child fall images using an adult-trained model. Among all epoch settings, the model performed best at 50 epochs, achieving a precision of 0.85 and a recall of 0.91. Although this performance was not quite as good as when they tested on adult data, it was shown that there was a high likelihood of detecting child falls even though it was not exposed to any child images that were used to train this.

These findings point to the fact that a YOLOv9 network trained on adult falls may be sufficiently accurate in children under the strict conditions of controlled environments, representing a solid starting point of safety systems in children at the early stages.

Table 2 *Performance comparison of the YOLOv9 model at different training epochs when tested on child fall images using an adult-trained model*

Metric	Epoch 30	Epoch 50	Epoch 70	Epoch 100
Precision	0.75	0.85	0.87	0.98
Recall	0.87	0.91	0.82	1.00
F1 Score	0.81	0.88	0.84	0.99
mAP@0.5	0.87	0.89	0.87	1.00
mAP@0.5:0.95	0.53	0.62	0.61	0.68

4.3 Investigation on the Performance of Child Fall Detection in Multi-Subject Frames Using an Adult Fall Detection Training Dataset

In this section, the author looks at the child fall detection performance of the YOLOv9 model in complex scenes or situations with multiple subjects in them. Although training the model used a dataset only of adult fall images with isolated people in the picture, the test images used in this experiment had other two or more children in a single frame. Such an arrangement was to test the resiliency of the model and its power to generalize in real-life settings, e.g. in school or in a playground, where objects have a tendency towards visual clutter and overlap in subjects. The example of the detection output shown in Fig. 15 demonstrates that the proposed YOLOv9 model can detect the child fall in a multi-subject scene without special training data despite the complexity of the appearance representation in that scene.

Fig. 16 presents the GUI output for a complex scene where multiple children are present, but no fall event occurs. The interface correctly displays the absence of any detection, which demonstrates the model's robustness in avoiding false positives even in cluttered environments with several moving subjects.

Fig. 17 shows the GUI detection output when a fall occurs in a multi-subject scene. Despite the increased complexity and visual clutter, the system successfully identifies the fall event and highlights it with a bounding box, underscoring its potential for use in real-world environments such as schools or childcare centers.

Fig. 15 *Detection output of a child fall in a multi-subject scene using an adult-trained YOLOv9 model*

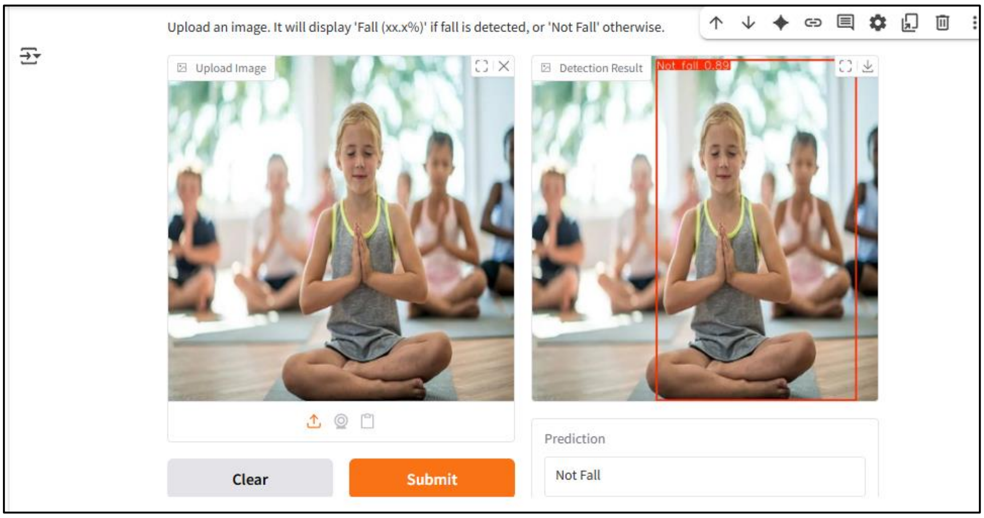

Fig. 16 *GUI output of multiple children in a non-fall scenario. The system does not generate false detections*

Fig. 17 *GUI output of child fall detection in a multi-subject scene. The system correctly detects and labels the fall despite the complex background*

Despite the increased complexity, the model demonstrated a reasonable ability to localize fall events. Table 3 shows the performance comparison of the YOLOv9 model at different training epochs when tested on multi-subject child fall images using an adult-trained dataset. However, due to the presence of multiple bodies and dynamic backgrounds, its performance declined compared to previous experiments. At 50 epochs, the model recorded a precision of 0.78, a recall of 0.81, and an F1-score of 0.79. These results suggest that while the model can still detect falls in crowded scenes, its accuracy is affected by occlusion and visual ambiguity, particularly since it was not trained on multi-subject data.

This experiment highlights the importance of using more diverse and context-rich datasets for training, especially when deploying fall detection systems in environments where multiple individuals may be present.

Table 3 *Performance comparison of the YOLOv9 model at different training epochs when tested on multi-subject child fall images using an adult-trained dataset*

Metric	Epoch 30	Epoch 50	Epoch 70	Epoch 100
Precision	0.768	0.896	0.922	0.916
Recall	0.750	0.833	0.984	0.906
F1 Score	0.759	0.863	0.952	0.911
mAP@0.5	0.840	0.928	0.989	0.969
mAP@0.5:0.95	0.565	0.614	0.695	0.720

In this study, the F1 Score was used as one of the key performance metrics to evaluate the effectiveness of the YOLOv9 model in detecting fall events. F1 Score is a composite measure that adds precision (the percentage of correctly detected falls in all detected falls), and recall (the percentage of actual falls that are indeed detected) to one score. It is determined by formula:

$$\text{F1} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{1}$$

This is an important metric when and where the occurring of both false positives (a non-fall identity as a fall), and false negatives (a missed fall event) are crucial. In the case of fall detection, this is particularly sensitive. The F1 Score therefore allowed of a middle ground in regard to accuracy with which the model detects the different test situations which involve adult-only, child, and multi- subject datasets in this study. With the application of the F1 Score, the research could have a better estimation of the reliability of the overall model and also in its application to child fall detection of the model even though it was only trained on the adult fall data.

4.4 Summary of YOLOv9 Fall Detection Performance Across Experiments

The three experiments collectively demonstrate the strengths and limitations of using an adult fall dataset to train a generalized fall detection model. Table 4 shows the comparison table summarizing all three experiments. In Experiment 1, the model showed strong performance when tested on adult fall images, achieving an average mAP@0.5 of 0.982. When applied to single-child fall images in Experiment 2, the model retained good generalization ability, though performance slightly declined (average mAP@0.5 = 0.908). Experiment 3 introduced further complexity with multi-child scenes, where the model maintained moderate accuracy (average mAP@0.5 = 0.932), despite challenges like clutter and overlapping subjects. These findings indicate that while adult fall data provides a solid foundation, child-specific datasets are crucial for improving detection performance in real-world environments.

Table 4 *Comparison table summarizing all three experiments*

Experiment	Dataset Size	Train / Valid / Test Split	Test Data Type	Epochs tested	Key Findings
1 (Task 1)	100 images	80 adult / 10 adult / 10 adult	Adult fall images	30, 50, 70, 100	Achieved the highest mAP@0.5 of 0.995 at epochs 50 and 100. Best overall performance with consistent precision and recall. Serves as the performance benchmark.
2 (Task 2)	110 images	88 adult / 11 adult / 11 child	Child fall images (single child in frame)		Showed partial generalization. mAP@0.5 reached 1.00 at epoch 100, but precision and confidence were lower than in Task 1.
3 (Task 3)	120 images	96 adult / 12 adult / 12 child	Child fall images (multiple children in frame)		Performance decreased in complex scenes. Highest F1 score at epoch 70 with mAP@0.5 of 0.989. Accuracy is affected by overlapping subjects.

5. Conclusion and Recommendations

This study successfully achieved its main objectives by developing and evaluating a fall detection system using the YOLOv9 deep learning model. The model showed strong performance when trained on adult fall images and tested across child scenarios, confirming its potential for generalization. The best results were observed at 50 training epochs, with high precision, recall, and mAP, especially in single-child cases. While performance decreased slightly in multi-subject scenes, the system still demonstrated practical reliability.

To improve the system's real-world applicability, several enhancements are recommended: (1) Expand the dataset to include real child fall data to improve accuracy, (2) Apply domain adaptation techniques to boost generalization, (3) Integrate the system with IoT devices for real-time alerts, and (4) Implement confidence-based alert thresholds to minimize false positives. These improvements would increase the model's robustness, safety, and suitability for deployment in environments such as schools or homes.

Acknowledgement

The authors would also like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Hazli Roslan, Nabilah Ibrahim;* ***data collection:*** *Nurul Khalidah Mohd Yusof;* ***analysis and interpretation of results:*** *Nurul Khalidah Mohd Yusof, Suhaila Sari, Nik Shahidah Afifi Md Taujuddin;* ***draft manuscript preparation:*** *Nurul Khalidah Mohd Yusof, Suhaila Sari. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] M. Rama Bai, J. Sreedevi, A. Korukonda, I. Sri, and R. Saketh, "and Communication Engineering Impact Factor 8.102∣□Peer-reviewed & Refereed journal∣□Vol," *IJARCCE International Journal of Advanced Research in Computer*, vol. 13, no. 5, 2024, doi: https://doi.org/10.17148/IJARCCE.2024.13549.

[2] "What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," Arxiv.org, 2015. https://arxiv.org/html/2409.07813v1

[3] Falcons-ai, "GitHub - Falcons-ai/fall_detection_notebook," *GitHub*, 2024. https://github.com/Falcons-ai/fall_detection_notebook

[4] Z. Yang, B. Tsui, J. Ning, and Z. Wu, "Falling Detection of Toddlers Based on Improved YOLOv8 Models," *Sensors*, vol. 24, no. 19, p. 6451, Oct. 2024, doi: https://doi.org/10.3390/s24196451.

[5] X. Huang *et al.*, "SDES-YOLO: A high-precision and lightweight model for fall detection in complex environments," *Scientific Reports*, vol. 15, no. 1, Jan. 2025, doi: https://doi.org/10.1038/s41598-025-86593-9.

[6] N. T. Newaz and E. Hanada, "The Methods of Fall Detection: A Literature Review," *Sensors*, vol. 23, no. 11, p. 5212, Jan. 2023, doi: https://doi.org/10.3390/s23115212.

[7] X. Wang, J. Ellul, and G. Azzopardi, "Elderly Fall Detection Systems: A Literature Survey," *Frontiers in Robotics and AI*, vol. 7, no. 71, Jun. 2020, doi: https://doi.org/10.3389/frobt.2020.00071.

[8] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems." Available: https://core.ac.uk/download/pdf/81788103.pdf

[9] "A Flexible Fall Detection Framework Based on Object Detection and Motion Analysis | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/document/10066990

[10] X. Wang and K. Jia, "Human Fall Detection Algorithm Based on YOLOv3," Jul. 2020, doi: https://doi.org/10.1109/icivc50857.2020.9177447.

[11] "An Image-based Fall Detection System for the Elderly using YOLOv5 | IEEE Conference Publication | IEEE Xplore," *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/document/10404248

[12] G. Team, "Quickstart," *gradio.app*. https://www.gradio.app/guides/quickstart

[13] H. Martinez, "Introduction to Gradio for Building Interactive Applications - PyImageSearch," *PyImageSearch*, Feb. 03, 2025. https://pyimagesearch.com/2025/02/03/introduction-to-gradio-for-building-interactive-applications/

[14] Bex Tuychiev, "Building User Interfaces For AI Applications with Gradio in Python," *Datacamp.com*, Jul. 15, 2024. https://www.datacamp.com/tutorial/gradio-python-tutorial

[15] M. Das, "Gradio: The New Frontier in Interactive Python Data Apps □," *Medium*, Apr. 08, 2023. https://web.archive.org/web/20231222020108/https:/medium.com/@HeCanThink/gradio-the-new-frontier-in-interactive-python-data-apps-64b5ce06628a