A Novel Computer Vision-Based Approach to Mitigating Fall Risks in the Elderly through Spatial-Channel Decoupled Downsampling in YOLOv10

Ajay Singh[1,*], Alok Katiyar[1]
[1]Galgotias University, Noida, India
Emails: ajaysingh@galgotiacollege.edu; alok.katiyar@galgotiasuniversity.edu.in

Abstract

Elderly health has always been a matter of concern for the medical doctors and researchers to come up with advanced recovery techniques. With the rise in population of elderly people and mostly residing alone at home in solitude has motivated many researchers to work on remedial measures for the biggest safety risk faced by them which is elderly fall prevention and mitigating thereby causes of injuries. In this paper, an intelligent deep learning and computer vision based elderly fall recognition system is designed which utilizes advanced spatial-channel decoupled downsampling in You Only Look Once version 10 (YOLOv10), pytorch, darknet and cascaded CNN technologies for the fall detection. The results after testing manifest that the accuracy of the proposed system to recognize and detect the elderly fall is quite assuring, the values of accuracy and mean Average Precision (mAP50) coming out to be 92.46% and 94.1% respectively after the model validation. Moreover, the system displays a real time performance as it can process approximately 45 frames of images per second that realizes a real-time identification of elderly fall patterns. As compared to previous models, the proposed model is much more efficient and has shown promising results.

Keywords: YOLOv10 Object Detection Algorithm; Computer Vision; Machine Learning; Image Processing; Spatial-channel decoupled down-sampling technique; Non-maximum suppression

1. Introduction

According to a recent World Health Organization research [1], falls rank as the second most common cause of fatal unintentional injury deaths throughout the world. Globally, an estimated 684000 persons pass away because of falls every year. Moreover, the biggest death rate from falls occurs in people 60 years of age and above. For older people over 75, falls are the most common cause of mortality, and it is the second most common mortality cause for persons whose age-range lies between 65 and 75 [2]. Therefore, research in the field of automatic fall detection is crucial for advancement of understanding and developing new solutions especially to ensure elder persons safety and injury avoidance as most of them live alone in solitude, all the other family members being busy in day-to-day schedules. An intelligent fall detection system keeps an eye on senior citizen housing and, in the event of a fall, it notifies caregivers or an emergency center with an alarm signal to avoid any serious consequences. To detect and prevent falls, numerous methods and algorithms have been developed so far. According to [3], fall detection systems are developed by following three main category approaches: computer vision based approach, detection with ambient devices and wearable sensor based approaches. Elderly persons are uncomfortable with ambient devices and sensors as these approaches involve the use of clumsy electronic equipment, which most elderly are not convenient with and avoid using so computer vision approaches fit better in this scenario as proposed in this system. In this work, using the cutting-edge technologies like spatial-channel decoupled downsampling, non- maximum suppression free object detection through YOLOv10 framework, the suggested methodology is designed to offer an efficient solution for elderly fall detection in all scenarios where any fall is detected in live video through webcam or input images. The main goal of this technology is to improve overall safety of elderly people by reducing reaction times of prevention to fall accidents.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

2. Related Work

Guang Cheng et al. [4] proposed a low-cost small mobile robot vision-based fall detection method where they used RPi 4, Model B as hardware and lightweight Nano Det-Lite object detection model. They used a 2.17 MB sized model and obtained a Mean Average Precision (mAP) of 0.192. The model was trained on a server having deep learning architecture with NVIDIA Quadro M4000 GPU and used Ubuntu 16.04 as IDE. RaspberryPi-4 Model B having 4GB internal memory was used as the robot. A camera was mounted on mobile robot and detection algorithm was deployed on mobile robot platform. Nano Det-Lite (optimized from NanoDet) can be deployed conveniently, is training-friendly and has the advantages of fast speed. They showed a high recall and precision in the results however this system was quite complex and difficult to monitor as the robot could not always adhere with the elderly people and many times elderly people are quite uncomfortable with this kind of environment. Vyshakh Krishnan T et al. [5] proposed a YOLOv8 system that is based on advanced deep learning and computer vision methods to provide efficient fall detection. The backbone of this system is an iteration of the cspdarknet53 model that has 53 convolutional layers. The training process was expedited by using Google Colab Pro, which has excellent GPU support. The PyTorch framework was used for model development due to its wide-ranging usefulness in deep learning research. The GeForce RTX 2080 Ti model of NVIDIA graphics processing unit (GPU) was used in the experiments on a personal computer. The model was already pre-trained on the COCO dataset. They compared YOLOv8 with YOLOv5 and YOLOv3 and found better accuracy in terms of mAP.

Kumar et al. [6] proposed an object detection system to help visually impaired persons. Objects were detected using camera used as the device to input videos, images in real time, and conveyed the presence of objects to the users using android-based smartphone connected to headphones. Yolo v3 object detection algorithm was used. Yolo v3 pretrained weights were used and coco dataset used by which each object is processed and labelled, then text to speech converter is used and the detected object is converted to speech from telling the object name to user however the system was not working on larger datasets and improvement was required through re-training on larger datasets. Niu et al. [7] proposed a system that combinedly used OpenCV library along with EAIDK-310 embedded development board. Multiparameter Comprehensive Evaluation Algorithm was used which is based upon 3 features-centroid height, human aspect ratio and inclination angle. With camera, they use ARM processor (Rui Xin-micro RK3228H) in their hardware structure. EAIDK-310 embedded board they used. In the event of fall detection, a remote alarm is sent via ethernet interface. Hybrid Gaussian model is used for Object Detection where the background pixels subjected to Gaussian distribution. EAIDK-310 embedded board ran the algorithm with the help of installed OpenCV library functions to extract the 3 contour features. They have compared EAIDK-310 with general-purpose computer detection results and proved better for modification. Pita et al. [8] proposed a Yolo v3 Machine Learning model along with Deep Transfer Learning Object Identification Approach. Anaconda IDE used, some GUI made for photos, live stream video and recorded video object detection. Various models were prepared in which Model-1 showed the lowest performance, meanwhile Model-2 to 20 showed improving results whereas Model 21 showed the highest performance. Mahesh & Kalidas in [9] have used an IoT-based fall detection and warning system that employs accelerometer and gyroscope sensors to continuously monitor the elderly fall detection. The setup employs a Node MCU microcontroller and an MPU6050 sensor module, which houses both an accelerometer and a gyroscope. The accelerometer provides critical three-axis data on angular parameters, while the gyroscope aids in orientation detection. In IFTTT, an applet is created with the event name set to "Fall Detected," and "Fall Detected" in both the alert's subject and message. Once a fall is detected, the system promptly notifies the individual via email.

Sudthongkhong et al. [10] employed a system that monitors for falls by identifying posture abnormalities using a dedicated camera and gesture recognition software and an advanced Image Processing for precise Pose Estimation. In order to determine the severity of a fall, Deep Neural Networks (CNN & RNN) evaluates Pose Estimation data, then severity of fall is determined and simultaneously caregivers are notified via the network so they can respond quickly. Camera feeds the data collection and Nvidia Jetson Nano Developer Kit serves as the hardware platform for data acquisition and processing. Jetson Nano collects sensor data from accelerometers and gyroscopes, adding the contextual information to the visual data. To analyse the captured images and videos, they use a combination of computer vision and image processing algorithms. Pose Estimation Algorithm & Fall Detection Algorithm were integrated with machine-learning algorithms. In Hardware Setup, Nvidia Jetson Nano Developer Kit, an ARM64-based computer for AI and data simulations is used along with a USB webcam, which serves as the primary image data source. The Tensorflow library of NVIDIA Jetson Nano provides a high computational power required for advanced computer vision and deep learning tasks. They achieved rapid detection due to high FPS (frames per second) achieved however the biggest drawback of this system is its complexity and looks to be quite expensive in terms of hardware usages. Xu et al. [11] employed a continuous motion detection technique based on radar called mmCMD using a dynamic feature visualisation (DFV) technique where the micro-Doppler matrix (MDM) elements are selectively mapped. By integrating into the YOLOv5 architecture, the coordinate attention (CA) and the specifically created fusion squeeze-and-excitation (FSE) module, a detection network is created for the visualized

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

micro-Doppler images. A DFV technique is used to selectively map significant values, emphasizing the motion of humans, in order to improve imaging quality. The FSE module and CA are combined with the YOLOv5 architecture to create a noval detection network known as YOLO-MDM, which incorporates global contextual information. At IoU thresholds ranging from 0.5 to 0.95, mmCMD has exhibited remarkable execution in detecting 12 motions, with an F1 score of 99% and an impressive mAP of 93%. It performed better than other widely used CNN-based and transformer-based models like YOLOX, DAB-DETR, and DINO with repect to both parameters, i.e., mAP metrics and inference time. Cheng et al. [12] presented a fall detection method based on YOLOv5-Lite considered as the lightweight object detection model. The main component of YOLO5-Lite is lighter than that of YOLOv5, and it uses shuffle blocks with shuffle channels and a truncated version of YOLOv5 head is used as the detection head. The model accuracy of YOLOv5-Lite is higher as comparable to YOLOv5 and it has a smaller parameter size as compared to YOLOv8. For labelling the images, they used makesense.ai website instead LabelImg tool as used by most authors previously mentioned. IDE/Computer configuration used are: CPU- Intel Core i5-10400F, 4000 MHz (40x100), RAM- 8GB, GPU- RTX3060(12G/ASUS), OS- Windows10, Softwares used are- PyCharm, python(version 3.7), CUDA (version 11.6), Pytorch (Framework 1.3) and used multiple cameras for taking photos of the person inside home to detect any fall. They achieved an accuracy of 92.1% and a good processing speed of 3.12ms. however the disadvantage in this system was the presence of false detections when it was a blurry image or the visibility was limited to only a small part of person's body.

Mundher & Jiaofei in [13] used a Kinect Sensor integrated with a mobile robot system that was used to monitor a person and detect whenever a fall occurs. Mobile robot has a connection with a smart phone that sends notifications as SMS and emergency call whenever a fall occurs. Kinect is a sensor that was inventedto be used an input device for the Xbox 360 gaming console and here it detected the speech recognition & gesture recognition by which elderly people can send commands to the robot. An event tracks the skeleton frame and whenever the elder person is detected, the processing of skeletal information gets started by the proposed algorithm. Some Threshold (Th) value of distance they take as the Th and if the Joint position < Th, the fallis detected; else it was a safe point. Robots keep themselves close to the bodies and raise alarm upon detecting a fall. Chan and Goh [14] proposed a computer vision-based system that can detect falls, store accident records, and deliver alarm messages. The system is based on real-time videos collected in indoor situations via a 2D RGB camera. MediaPipe Holistic was used to apply posture estimation to extract the appearance feature in terms of old people's skeleton important points, from the input videos. Next, the long short-term memory (LSTM) deep learning model receives this feature as input. UR Fall Detection (URFD) dataset is used for evaluation. The framework achieved an accuracy of 87% in fall detection, 77% in choking detection, and 83% in Activities of Daily Living detection. MediaPipe Holistic processes live video streams to extract crucial information for senior citizens, emphasizing hand landmarks and posture. The input data (each wouth 30 frames with 258 keypoints coordinates) is used to produce the feature array "X" and label array "y". TensorFlow and Keras are utilized in building LSTM neural networks. Loss function is employed with the help of adam optimizer, and categorical crossentropy. After training and testing, the framework exhibits a promising result that shows a better detection accuracy. Roboflow website (elderly fall detection dataset) [5], fall dataset from Kaggle [15], URFD dataset [16] have been used for the collection training and testing images data in most of the previous works. Lin et al. [17] employed an edge computing technique based on artificial intelligence to identify falls. The VGG16 computer vision model was utilised by Chhetri et al. [18] to identify falls incorporating the transfer learning methods. Fei et al. [19] suggested a two-stream network in a different study for one-person fall detection in 2D RGB videos. RAFT network is used for optical flow to extract motion features, and OpenPose is used for human pose estimation to extract appearance features. After that, the two features are combined and fed into a classifier to classify binary falls. In the trial, the suggested method's accuracy was 100% on the URFD dataset and 98.95% on the Le2i dataset. The approach demonstrated good performance in various scenarios and dimly lit areas. Furthermore, the ablation study discovered that the optimal performance is achieved when the authors combine RGB, optical flow, frame difference, and human pose for the verification of fall detection. Pre-fusion performed better than post-fusion for feature fusion of the two-stream network of optical flow and human pose. While post-fusion employs features independently to generate two results from two streams and takes the union of positive samples for classification, pre-fusion fuses the features prior to classification. De et al. [20] described an approach that combines human shape-based and motion-based features to detect falls in the elderly population. Motion History Images (MHI) are used to represent temporal features, and the centroid and height-to- width ratio (HWR) of the moving subject are examples of spatial features. The classification model uses both threshold-based and keyframe-based techniques in a two-channel manner. For precision, these channels are integrated based on classification disparities and additional data. Keyframes are chosen and then classified using K-NN based on the spatial feature displacement exceeding a predetermined threshold. The suggested algorithm shows encouraging outcomes in daily activities and simulated falls.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

3. Proposed Architecture

YOLOv10, invented by Tsinghua University researchers using the Ultralytics Python package is the most recent iteration of the YOLO (You Only Look Once) object detection series. The newest model in the YOLO family of object detection models, which is renowned for its real-time object identification capabilities, is called YOLOv10. YOLOv10 builds on the success of its predecessors by pushing the boundaries of performance-efficiency. Real-time object detection in a variety of applications is expected to be revolutionized by the new and interesting improvements. Scholars have carried out a great deal of research with the YOLO models, making significant advancements to come up with YOLOv10 model.

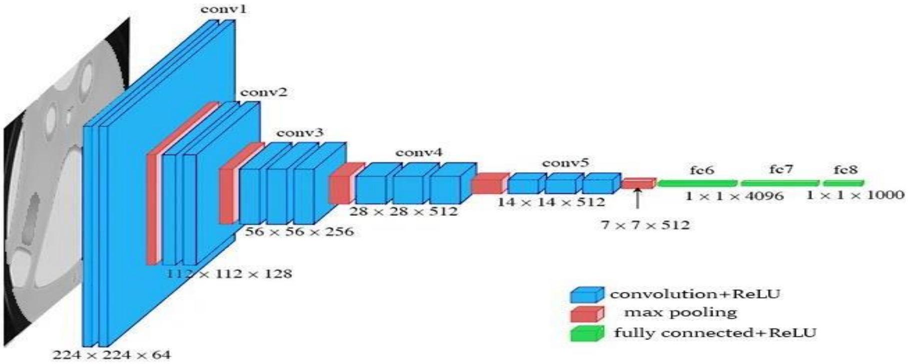

Figure 1. Convolution Process in YOLOv10

YOLOv10 is a highly sophisticated and advanced computer vision model and has introduced numerous major advances to boost efficiency and performance in addi-tion to the convolution process as shown Fig.1. Real-time end-to-end object identifi-cation has achieved a new dimension with Yolo10, which fixes the post-processing and model design flaws as in earlier YOLO versions. The advanced features of YOLOv10 include NMS (Non-Maximum Suppression)-free training, lightweight clas-sification heads, and spatial-channel decoupled down-sampling and rank-guided block design. NMS that is used mostly in previous versions of YOLO is computationally expensive and requires high inference time. Spatial-channel decoupled downsampling provides pointwise (1x1 filter) to add on the number of channels and depth wise convolution (3x3 convolution with Stride of 2) to decrease the spatial dimensions as shown Fig.2.

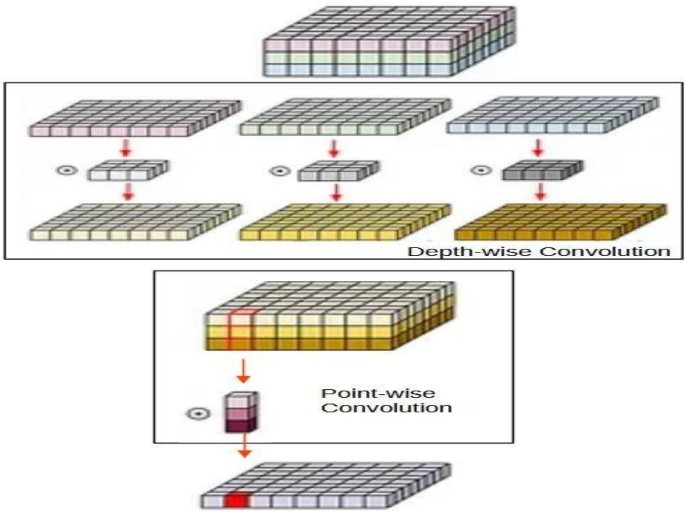

Figure 2. Depthwise and Pointwise Convolution

Due to this decoupling, the computational cost of the system is decreased to $O\,(2HWC^2 + \frac{9}{2}HWC)$

from $O\,(^{9}\,HWC^2)$ and the reduced param counts are $O(2C^2 + 18C)$ which leads to increased information retention and reduced latency. In rank-guided block design, all the convolution stages are sorted based on their

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

intrinsic ranks in ascending order. The basic blocks in stages with higher redundancy (lower rank) are replaced with the more efficient compact inverted block (CIB) structure. YOLOv10's architecture incorporates a number of significant advances while enhancing the features of earlier YOLO versions [21]. The following elements make up the YOLOv10 architecture: 1. Backbone: which is in charge of feature extraction, employs a refined CSPNet (Cross Stage Partial Network) to lessen computational redundancy and boost gradient flow. 2. Neck: created to transfer characteristics to head by combining them from several scales. For efficient multiscale feature fusion, PAN (Path Aggregation Network) layers are included. 3. One-to-Many Head: produces many predictions for each item in the training process to enhance learning accuracy and offer efficient supervisory signals. 4. One-to-One Head: reduces latency and boosts efficiency by creating a single best prediction for each object during inference, doing away with the requirement for NMS. In addition, in this model a consistent matching metric is implemented to improve the dual label assignments of YOLOv10. Through the implementation of this metric, the alignment of both one-to- one and one-to-many heads is achieved during their training. This consistent matching metric can be defined as

$$m(\alpha, \beta) = s \, . \, p^{\alpha} \, . \, IoU(\epsilon, b)^{\beta} \text{ (Fig. 3)} \quad (1)$$ where,

p denotes *the classification score,*

ϵ and b denotes *bounding boxes for prediction and instance respectively*

s denotes *if the prediction's anchor point is within the instance*

The importance of classification and localization tasks are balanced by the parameters α and β, where:

$$m_{o2m} = m(\alpha_{o2m}, \beta_{o2m}) \quad (2)$$

$$m_{o2o} = m(\alpha_{o2o}, \beta_{o2o}) \quad (3)$$

Here, Equation (2) denotes denote one-to-many metrics Equation (3) denotes one-to-one metrics.

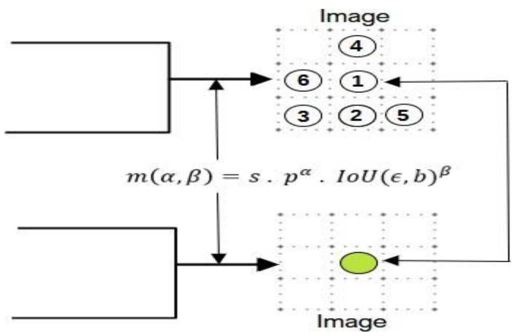

Figure 3. Matching Metric workflow

Video inputs are supplied into the YOLOv10 model, and each frame is preprocessed in accordance with the Yolov10 input format before being divided into frames. Whenever the model identifies any fall cases in the frame, it will forecast the output as bounding box values. Fig.3 shows the workflow of the proposed system.

4. Methodology

The rigorous and laborious methodology included collection and preparation of data, preprocessing, model adjustment, train-test splitting and finally model training and evaluation. For training, dataset was created by collecting and combining data from multiple sources- elderly fall dataset from roboflow website [23], UR fall detection dataset [24-25] and the fall dataset from Kaggle [26]. Besides, some real-world fall scenarios from senior citizen homes were also collected. Datasets were annotated with the help of bounding boxes drawn that could bound persons (elders) in the images. 70 percent of data was taken for training and 20 percent for testing and 10 percent for validation and the data splitting was done in such a way that both fall and non-fall cases of images are equally distributed. The dataset created by combining the fall datasets from Kaggle, Roboflow and UR dataset are taken and provided as custom dataset to train the model. During the preprocessing part, all the images are processed through the labelImg software to obtain the text files containing the classes and coordinates of the respective images. All the images are then reduced to the default size of YOLO models- (640,640,3) by using the Min-max normalization. The Gaussian Filter used with YOLOv10 converts the input image into a 13x13 grids to improve the accuracy of detection for multiple scales.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

Figure 4. Sample Input Images used for Training purposes

Flipping and translation were implemented as the data augmentation techniques during training process so that the model is tuned accordingly to various scenarios. However, the YOLOv10 model is quite robust but following modifications were implemented in this work to achieve an improved accuracy- the anchor box's dimensions were increased, the hyper parameters were fine-tuned and the output layer of the model was also modified. Then we fine-tuned our trained YOLOv10 model on a custom. Data set. YOLOv10 is the latest state of the art real time object detection model that outperforms all the other YOLO models in terms of speed, accuracy and efficiency parameters. YOLOv10 model takes less time to do object detection on input image unlike other YOLO models that take comparatively large amount of time to process an input image, which can be seen in our model. Therefore, the inference latency of YOLOv10 model is very good as compared to other YOLO models. YOLOv10 adopt efficiency driven design strategy, which reduces the computational overhead and is optimized at various model components as well. Training on all input images is done after fine-tuning the model weights. The fine-tuning of YOLOv10 model and training the YOLOv10 model is done on Elderly Fall Detection data set that was taken from Roboflow website combined with other input datasets. Therefore, we will have two different classes in our data set, which includes falling and non-falling elderly. In addition, our data set is very much balanced in terms of dataset throughput. Firstly, the data set is exported into Google Colab notebook. YOLOv10 is cloned through the GitHub repository [22] and all the required ultralytics packages are installed that are required to fine-tune the model. The pre- trained model weights are downloaded and all the YOLOv10 model weights can be found in 'weights' folder. Then we train the model on the custom-dataset after tuning the batch size and epochs required for the model training. Model training is done for 100 epochs on elderly fall-detection data set. High precision score is obtained which means the number of false-positives were quite less. Several parameter metrics such as the precision, accuracy, recall, F1 score, and mean average precision (mAP) were used to evaluate the model's performance on a validation data set. YOLOv3, YOLOv5 and YOLOv8 were the baseline models utilized in the comparison analysis (Table 1).

Figure 5. Workflow of the Proposed Model

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

Moreover, a personal computer with an Intel(R) Core(TM) i7-8565U CPU @ 1.99 GHz processor and NVIDIA graphics processing unit (GPU) was used for the experiments. Google Colaboratory with an extensive GPU support was used. The PyTorch framework was selected for the model's development in spatial-channel decoupled downsampling and YOLOv10 algorithm.

5. Experiments & Results

Our model consisted of 385 layers, 2707820 parameters, 2707804 gradients and 8.4 GFLOPs and this YOLOv10 architecture based fall-detection system has demonstrated extremely promising results in precisely identifying the falls. The ability to detect and classify fall instances accurately can be assessed with the help of mean average precision (mAP) which the system has achieved quite remarkably with 94.1% when the intersection over union (IOU) varies from 0.5 to 0.95. The system identified falls with very less number of false positive occurrences being another virtue of the system along with high precision and recall.

Table 1: Comparison of different YOLO models using the same dataset in 100 epochs

S. No.	MODEL	ACCURACY (%)	MAP50 (%)
1	YOLOv10 (This proposed model)	92.4	94.1
2	YOLOv8	89.6	90.1
3	YOLOv5	77	75.5
4	YOLOv3	76.5	71.8

A detailed evaluation of the model's assessment was carried out by the computation of metrics such as precision, recall, and F1 score metrics. The recall metric reflects the system's ability to identify a substantial number of actual fall incidents while high precision score showed the system's capability to predict the falls accurately. More proof of system dependability as a fall detection system is achieved through the F1 score (accounting for both recall and precision) which is quite balanced in the results.

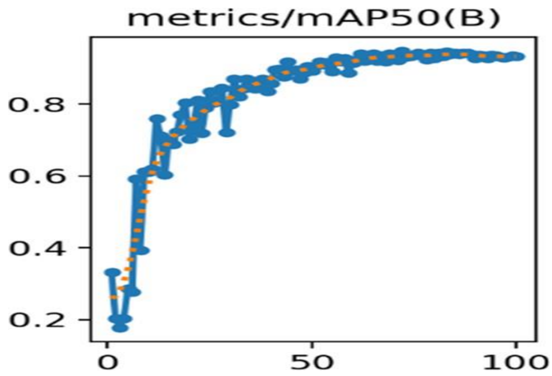

Figure 6. values of mAP50 in epochs from 1-100

The mean average precision graph in Fig.6 illustrates the system’s performance with respect to mean Average Precision (mAP) when using a 50% intersection over union (IoU) threshold used for the measure of bounding box (B). At the x-axis, we have an indeterminate variable that is presumed in model experiment settings which scales from 0 to 100 and mAP50 (B) score is shown by the y-axis which is scaling from 0.2 to 0.10. Positive correlation is depicted by this graph with a linear plot and having a direct proportionate relationship between x-axis variable and the mAP50 (B) score. When the x-axis variable reaches its maximum value, the system achieves a mean average precision (mAP50 (B)) score of approximately 0.94 at 50% intersection over union.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

Table 2: Comparison of various metrics in different epochs

S. No.	Epochs	mAP50	box loss	cls loss
1	10	61	1.387	1.89
2	25	83.5	1.195	1.25
3	50	89.3	1.027	0.89
4	75	93.7	0.893	0.68
5	100	94.1	0.460	0.28

TABLE 2 illustrates the values of various metrics achieved during the training process where the mean Average Precision persistently increases with the increase in number of training epochs and the highest point is reached at 94.10% at the 100th epoch. Both the classification and bounding box validation losses are decreased with epoch's increment, which clearly shows that the model depicts enhanced ability to detect unfamiliar data and obtains enhanced precision within bounding boxes for the localization of falls patterns. All these metrics patterns show how well the model is able to identify and pinpoint human falls.

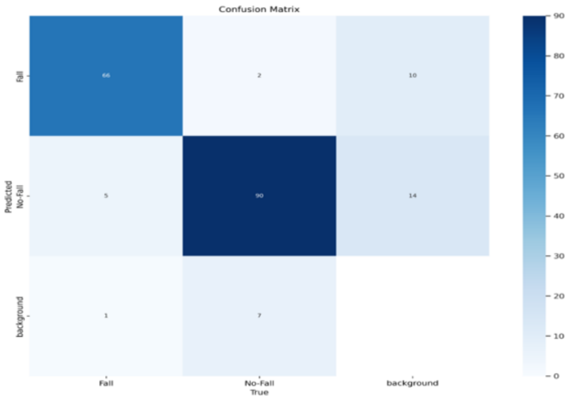

Figure 7. Confusion Matrix for the prediction

As shown in Fig.7, the model predicted approximately 66 instances of fall occurences. The value of false positives is very less here.

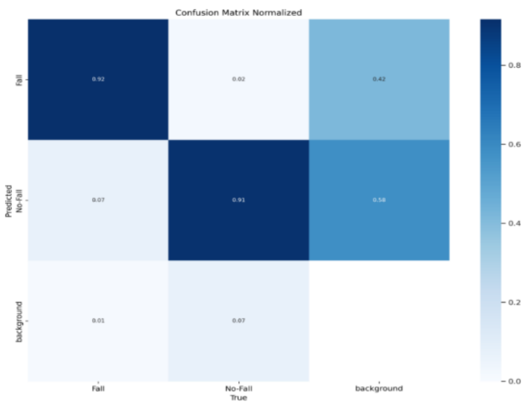

Figure 8. Confusion Matrix Normalized

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

Fig. 8 shows the normalized scores. 92% of the times when the elderly person was detected as falling, the model predicted correctly while 1% times the system was unable to detect anything.

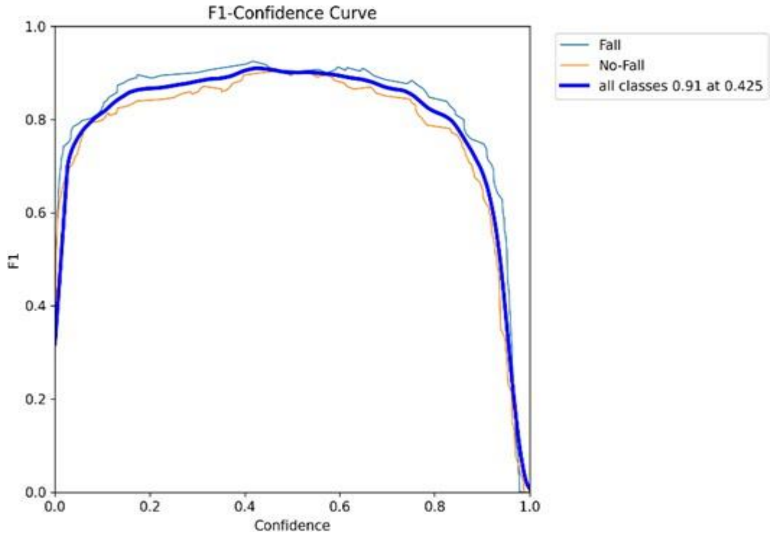

Figure 9. F1-Confidence Curve

Fig.9 shows the F1 confidence curve, which tells us the tradeoff between the precision and recall scores. The precision-recall curve shown in Fig.10 shows the core performance of the proposed elderly fall detection system. A critical equilibrium has been achieved by the precision-recall curve in between the precision i.e., the occurrences of incorrect alerts are reducing and the recall i.e., the occurrence of falls that are not detected also reducing. This trade-off is successfully balanced by the curve, yielding an impressive Area Under the Curve (AUC) value of 0.965. This high score illustrates the system's ability to accurately detect falls and minimize false positives.

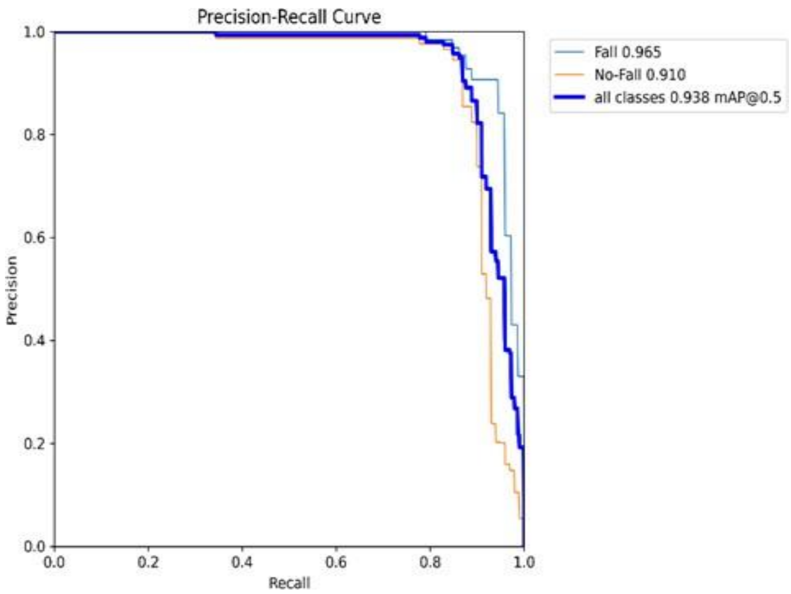

Figure 10. Precision-Recall Curve

Furthermore, the focus on the "Fall-Detected" category highlights how well the sys-tem detects falls in specific situations. The mean average precision (mAP) at an inter-section-over-union (IoU) threshold of 0.5 correlates exactly with the area under the curve (AUC) of 0.965 and illustrates enhanced precision. This suggests steady performance at different IoU levels. The capacity of this proposed elderly fall detection system is accurately and succinctly represented by this analysis. It effectively strikes a balance between memory and precision, leading to a high level of accuracy in fall detection and a decrease in false alarms.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

Figure 11. Model Predictions on Validation Dataset

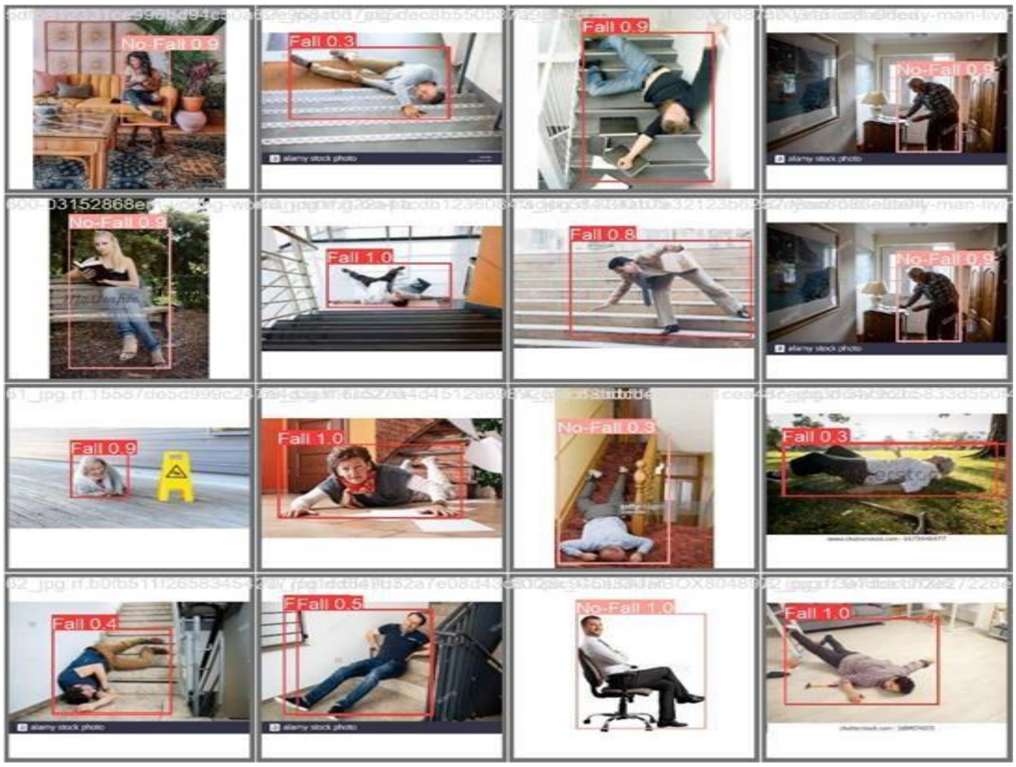

Figure 12. Model Predictions on Validation Dataset Fig. 11 and Fig. 12 shows model predictions on different validation datasets.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

Figure 13. Test Results on an input Image

Figure 14. Screengrab of Test Results on live video

The jupyter notebook was used for the purpose of elderly fall detection in any live video webcam. Model was placed at the correct location by providing the path to trained model from colaboratory. In addition, the live video captured through the webcam is identified for elderly fall detection as shown in Fig. 14. Fig. 13 shows the test results of an input test image.

6. Conclusion

In conclusion, the study presented a robust and precise fall detection system that makes use of the most recent and reliable YOLOv10 object detection framework. The system has demonstrated an amazing 94.1% accuracy (mAP) in a variety of fall scenarios, especially in assisted living facilities. Yolov10 self-attention and multiskilling technology has shown itself to be a good fit for real-time fall detection, meeting the urgent need to intervene promptly in cases of elderly falls. The methodology comprised an extensive process of data gathering, preprocessing the data, customizing the model, training and finally assessing the model. After modifying the output layer and anchor box dimensions and developing a special loss function to address the class imbalance, the Yolov10 model was optimized for fall detection. When comparing the experimental results to baseline models like YOLOv3 and YOLOv5, and YOLOv8 the suggested methodology showed a superior performance (mean Average Precision of 94.1%) compared to the other models. The model effectively and more- accurately detected falls, minimizing false positives and offering a dependable method of fall detection in senior care settings. In future work, a posture-detecting model can be implemented in the existing YOLOv10 framework that will provide a much more detail-oriented information about the pre and post-fall postures during the occurrence of a fall event so that assistance that is more precise could be provided to the falling individual.

Funding: "This research received no external funding"

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

References

[1] World Health Organization, "Falls," 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls.

[2] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, "Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor networks: A survey," *InTech eBooks*, 2010. [Online]. Available: https://doi.org/10.5772/13802.

[3] Y. Liu, P. C. K. Hung, F. Iqbal, and B. C. M. Fung, "Automatic fall risk detection based on imbalanced data," *IEEE Access*, vol. 9, pp. 163594–163611, 2021. https://doi.org/10.1109/access.2021.3133297.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

[4] G. Chen and X. Duan, "Vision-based elderly fall detection algorithm for mobile robots," in *Proc. 2021 IEEE 4th Int. Conf. Electronics Technology (ICET)*, 2021. https://doi.org/10.1109/icet51757.2021.9450950.

[5] T. V. Krishnan, B. K. Abhilash, and S. Govind, "A robust fall detection system for elderly persons using YOLO," in *Proc. 2024 IEEE 5th Int. Conf. Innovative Trends in Information Technology (ICITIIT)*, 2024. https://doi.org/10.1109/icitiit61487.2024.10580626.

[6] D. R. Kumar, H. K. Thakkar, S. Merugu, V. K. Gunjan, and S. K. Gupta, "Object detection system for visually impaired persons using smartphones," in *Lecture Notes in Electrical Engineering*, IEEE, 2021, pp. 1631–1642. https://doi.org/10.1007/978-981-16-3690-5_154.

[7] Y. Niu, "Research on elderly fall prevention recognition system for intelligent elderly care and machine vision," in *Proc. 2023 IEEE Int. Symp. Intelligent Robotics and Systems (ISoIRS)*, 2023. https://doi.org/10.1109/isoirs59890.2023.00033.

[8] M. S. U. Pita, A. S. Alon, P. M. B. Melo, R. M. Hernandez, and A. I. Magboo, "Indoor human fall detection using data augmentation-assisted transfer learning in an aging population for smart homecare: A deep convolutional neural network approach," in *Proc. 2021 IEEE 19th Student Conf. Research and Development (SCOReD)*, 2021. https://doi.org/10.1109/scored53546.2021.9652769.

[9] G. Mahesh and M. Kalidas, "A real-time IoT-based fall detection and alert system for elderly," in *Proc. 2023 IEEE Int. Conf. Advances in Computation, Communication and Information Technology (ICAICCIT)*, 2023. https://doi.org/10.1109/icaiccit60255.2023.10465914.

[10] C. Sudthongkhong, S. Suksri, C. Ratanaubol, S. Tepthong, J. Jitsupa, and P. Suksai, "NVIDIA Jetson Nano and Python-based economical human fall detection and analysis system," in *Proc. 2023 17th Int. Conf. Signal-Image Technology & Internet-Based Systems (SITIS)*, 2023. https://doi.org/10.1109/sitis61268.2023.00083.

[11] Z. Xu et al., "MMCMD: Continuous motion detection from visualized radar micro-Doppler signatures using visual object detection techniques," *IEEE Sensors J.*, vol. 24, no. 3, pp. 3394–3405, 2024. https://doi.org/10.1109/jsen.2023.3342103.

[12] B. Cheng, Y. Su, and Y. Cai, "Research on real-time human fall detection method based on YOLOv5-Lite," in *Proc. 2023 IEEE 6th Int. Conf. Electronic Information and Communication Technology (ICEICT)*, 2023. https://doi.org/10.1109/iceict57916.2023.10245195.

[13] Z. A. Mundher and J. Zhang, "A real-time fall detection system in elderly care using mobile robots and Kinect sensor," *Int. J. Materials, Mechanics and Manufacturing*, vol. 2, no. 2, pp. 133–138, 2014. https://doi.org/10.7763/ijmmm.2014.v2.115.

[14] X. E. Chan and C. P. Goh, "Computer vision-based home accident detection and response system for elderly care," in *Proc. 2024 IEEE 3rd Int. Conf. Digital Transformation and Applications (ICDXA)*, 2024. https://doi.org/10.1109/icdxa61007.2024.10470602.

[15] U. K. Kandagatla, "Fall detection dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/uttejkumarkandagatla/fall-detection-dataset.

[16] H. Xiao et al., "Towards privacy-supporting fall detection via deep unsupervised RGB2Depth adaptation," *arXiv preprint*, arXiv: 2308.12049, Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.12049.

[17] B. Lin et al., "Fall detection system with artificial intelligence-based edge computing," *IEEE Access*, vol. 10, pp. 4328–4339, 2022. https://doi.org/10.1109/access.2021.3140164.

[18] S. Chhetri et al., "Deep learning for vision-based fall detection system: Enhanced optical dynamic flow," *Computational Intelligence*, vol. 37, no. 1, pp. 578–595, 2020. https://doi.org/10.1111/coin.12428.

[19] K. Fei et al., "Flow-pose net: An effective two-stream network for fall detection," *The Visual Computer*, vol. 39, no. 6, pp. 2305–2320, 2022. https://doi.org/10.1007/s00371-022-02416-2.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025

[20] A. De, A. Saha, P. Kumar, and G. Pal, "Fall detection method based on spatio-temporal feature fusion using combined two-channel classification," *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 26081–26100, 2022. https://doi.org/10.1007/s11042-022-11914-3.

[21] A. Wang et al., "YOLOV10: Real-time end-to-end object detection," *arXiv preprint*, arXiv: 2405.14458, May 2024. https://doi.org/10.48550/arxiv.2405.14458.

[22] THU-MIG, "YOLOV10 GitHub repository." [Online]. Available: https://github.com/THU-MIG/yolov10.git.

[23] "Elderly fall detection dataset," Roboflow. [Online]. Available: https://universe.roboflow.com/fyp-sujy2/elderly-fall-detection/dataset/5.

[24] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed*, vol. 117, no. 3, pp. 489–501, 2014. https://doi.org/10.1016/j.cmpb.2014.09.005.

[25] M. Kepski, "UR Fall Detection Dataset." [Online]. Available: http://fenix.ur.edu.pl/~mkepski/ds/uf.html.

[26] "Elderly fall prediction and detection dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/laavanya/elderly-fall-prediction-and-detection.

DOI: https://doi.org/10.54216/FPA.190219
Received: January 19, 2025 Revised: February 16, 2025 Accepted: March 06, 2025