Course Project Report

Optimized YOLOv11 for Fallen person detection

Submitted By

Vinayaka S N (211AI040)
Vivek Vital Biragoni (211AI041)
Raghavendra (211AI029)
Venkatesh B M (211AI039)

as part of the requirements of the course

AI in Healthcare [Jul-Nov 2024]

in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology in Artificial Intelligence

under the guidance of

Dr. Sowmya Kamath S, Dept of IT, NITK Surathkal

undergone at

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

JUL-NOV 2024

DEPARTMENT OF INFORMATION TECHNOLOGY

National Institute of Technology Karnataka, Surathkal

C E R T I F I C A T E

This is to certify that the Course project Work Report entitled **"Optimized YOLOv11 for Fallen person detection"** is submitted by the group mentioned below -

Details of Project Group UPDATE TEAM MEMBER DETAILS

Name of the Student	Register No.	Signature with Date
Vinayaka S N	211AI040	
Vivek Vittal Biragoni	211AI041	
Raghavendra	211AI029	
Venkatesh B M	211AI039	

this report is a record of the work carried out by them as part of the course **AI in Healthcare (IT469)** during the semester **Jul-Nov 2024**. It is accepted as the Course Project Report submission in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Artificial Intelligence.**

(Name and Signature of Course Instructor)
Dr. Sowmya Kamath S.
Associate Professor
Dept. of IT, NITK Surathkal

D E C L A R A T I O N

We hereby declare that the project report entitled **"Optimized YOLOv11 for Fallen person detection"** submitted by us for the course **AI in Healthcare (IT469)** during the semester **Jul-Nov 2024**, as part of the partial course requirements for the award of the degree of Bachelor of Technology in Artificial Intelligence at NITK Surathkal is our original work. We declare that the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles elsewhere.

Details of Project Group UPDATE TEAM MEMBER DETAILS

Name of the Student	Register No.	Signature with Date
1. Vinayaka S N	211AI040	
2. Vivek Vittal Biragoni	211AI041	
3. Raghavendra	211AI029	
4. Venkatesh B M	211AI039	

Place: NITK, Surathkal
Date: 23/11/2024

Optimized YOLOv11 for Fallen person detection

Vinayaka S N[1], Vivek Vittal Biragoni[2], Raghavendra[3], Venkatesh B M[4]

***Abstract*— In the medical field, fall detection is essential, especially for emergency response systems and senior care. In order to improve contextual understanding of a scene, this study presents a unique framework that combines Convolutional Block Attention Modules (CBAM) with a modified YOLO11 architecture. The model concentrates on spatial and channel-wise crucial regions by carefully integrating CBAM at critical periods, allowing for precise distinction between fall and no-fall scenarios. Better feature refinement and contextual awareness are guaranteed by the YOLO11 improvements, outperforming more traditional systems like YOLOv8. Using Gradio(Gradio, 2024), the trained model was implemented in an intuitive application that offered an interactive online interface for fall detection in real time. After uploading and processing video, users can examine the findings, which emphasize fall-related incidents. When a fall is detected, the device also has a real-time warning mechanism that notifies caretakers or other pertinent personnel via Telegram, guaranteeing prompt action. The framework's potential as a scalable and effective fall detection solution in dynamic healthcare environments is shown by this combination of deployment features and architectural enhancements.**

Keywords: **Fall Detection, YOLOv11**

EDIT LINK to Overleaf project: `https://www.overleaf.com/9759341212wvpbzkxjpnds#d90ab1`

I. Introduction

Particularly for vulnerable groups including the elderly, children with impairments, and patients with neurological or cardiovascular diseases, who are more likely to experience injuries from falls, falls pose a major health risk. Effective fall detection systems can guarantee timely interventions, but many of the existing approaches have significant flaws. The discomfort and intrusiveness of wearable sensors and other conventional fall detection methods can reduce user compliance. Moreover, these methods often need extremely complex equipment that may not be feasible for widespread usage. A non-invasive substitute is provided by computer vision (CV)-based methods, which use video data to track human activity and identify falls. Despite its promise, real-time performance is a problem for existing CV-based FD systems, particularly in complicated settings with occlusions or inadequate lighting. Furthermore, these systems' capacity to generalize successfully across a range of circumstances is hampered by the absence of sizable, well-annotated datasets.

New opportunities for improving FD systems have been made possible by recent developments in deep learning, especially with regard to cutting-edge object recognition models like YOLO (You Only Look Once) (Ultralytics, 2024). The design of YOLOv8 is optimized in this study to differentiate between "fall" and "no-fall" occurrences. By evaluating human paths in real time, integration with sophisticated tracking algorithms like ByteTrack and BoT-SORT improves the system's predictive capabilities. The system can predict possible falls by computing acceleration and velocity parameters, allowing for preventative measures.

By streamlining video processing through keyframe selection techniques and integrating Convolutional Block Attention Modules (CBAMs) into the YOLOv8 architecture(LearnOpenCV, 2024), the research significantly enhances feature extraction. The robustness and usability of the suggested method are ensured by addressing ethical issues related to dataset gathering and usage. Practical deployment issues are also examined, such as modifying visualization techniques for settings devoid of graphical user interfaces. Through the creation of an effective, real-time FD system that can manage challenging real-world situations, this project seeks to advance AI-driven healthcare. The suggested architecture offers a major step toward dependable and flexible fall detection solutions by combining state-of-the-art tracking, analytical methods, and predictive modeling.

II. Literature Survey

A novel method for real-time fall detection (FD) employing an improved version of YOLOv8 is presented in the paper "Visionary Vigilance: Optimized YOLOv8 for Fallen Person Detection with Large-Scale Benchmark Dataset" (Alam et al., 2022). The work intends to solve the main drawbacks of current FD systems, such as their restricted real-time performance, computational inefficiency, and lack of large-scale, diversified datasets. In order to address these issues, the authors present a large dataset of 10,500 annotated photos that capture a variety of situations, including different lighting conditions, fall angles, and indoor and outdoor settings. This dataset improves the robustness and generality of FD systems, offering a solid basis for further study in the field.

Significant improvements to the YOLOv8S model are incorporated into the suggested system, such as convolutional block attention modules (CBAMs) at key points to fine-tune spatial and channel contexts and a focus module for effective spatial feature extraction. Extensive benchmarking against 13 state-of-the-art (SOTA) methodologies validates the improved detection accuracy and computing efficiency that these innovations produce. The model's dominance in complicated scenarios is demonstrated by metrics like mAP, precision, recall, and F1-score. The project advances real-time FD systems and establishes a new standard in intelligent visual surveillance by making the dataset and code publically available.

The goal of Ekram Alam et al.'s (Khan et al., 2024) "Vision-based human fall detection systems using deep learning: A review" is to present a thorough analysis of current developments in non-intrusive, vision-based, deep learning (DL)-based HFD systems. This article focuses exclusively on DL techniques, highlighting their benefits over traditional approaches in terms of automation, generalization, and performance, in contrast to earlier studies that cover a wider range of HFD methods. Its systematic examination of DL designs, performance indicators, and benchmark datasets while spotting deployment gaps in the actual world is what makes it distinctive. The approach comprises a thorough analysis of the literature published since 2014, classifying approaches according to DL models such as CNNs, LSTMs, and hybrid approaches, and assessing datasets and metrics including accuracy and sensitivity. The findings address issues like data-hungriness and processing demands while highlighting the greater accuracy and versatility of DL-based approaches, offering suggestions for further study and real-world implementation.

(Yu et al., 2022) uses a combination of Online Extreme Learning Machine (OS-ELM) and Federated Learning (FL) to tackle the problem of detecting falls in older people. In contrast to conventional methods that employ data from youth, which results in lower accuracy because of variations in movement patterns, this strategy uses FL to protect data privacy while facilitating user-to-user collaborative model changes. Its use of FL to improve generalizability without sacrificing privacy and OS-ELM to modify settings based on user-specific misclassified data is what makes it distinctive. Following the extraction of data including acceleration magnitude and fall stages, a trained ELM model is locally deployed and iteratively improved via OS-ELM and federated aggregation. Findings indicate that the Fed-ELM outperforms current techniques in terms of accuracy, sensitivity, and specificity, especially when it comes to detecting falls in the elderly, where accuracy surpasses 96%. This method highlights areas for future advancements in real-world circumstances while striking a balance between user-specific adaptability and generic resilience.

III. Problem Statement

Existing fall detection methods are often intrusive and ineffective. Our project aims to develop a real-time, accurate, non-intrusive and user-friendly fall detection system using an enhanced version of YOLOv8.

A. *Objectives*

- Create a system that can detect falls in real-time from both live video feeds and locally uploaded videos.
- Integrate advanced techniques, such as convolutional block attention modules (CBAMs), with YOLOv8 or better to improve the accuracy of fall detection in complex environments.
- Implement a dashboard that provides an intuitive interface for users to view detection results and manage video uploads.

IV. Dataset Overview

About 10,500 annotated samples, including both "Fall" and "No Fall" cases, make up the dataset used for fall detection. Bounding box annotations with normalized areas and center coordinates are included with every sample, giving each detected instance exact size and spatial information. These comments are critical for assuring consistency and correctness in training and evaluation. The dataset is also organized to facilitate easy interaction with frameworks for detection and recognition, which simplifies its use in practical applications. The dataset's diversity, which captures a range of environmental circumstances and situations that improve the detection models' resilience and generalizability, is one of its main strengths.

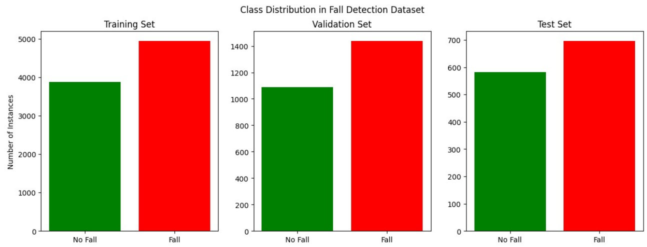


Fig. 1: Class distribution in fall detection dataset

The Training, Validation, and Test sets all show that most occurrences are concentrated in smaller normalized areas, according to the bounding box size distribution study. An evenly distributed dataset is reflected in this uniform distribution across subsets. The overlap in size distributions between the "Fall" and "No Fall" groups, however, is an important finding that suggests size might not be a key differentiator. This emphasizes how sophisticated feature extraction methods are required to reliably distinguish between the two classes.

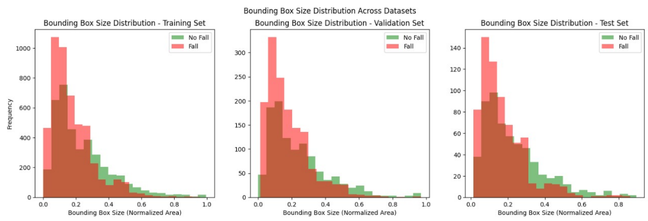


Fig. 2: Bounding box size distribution

The features of the dataset are further clarified by the bounding box center distribution. Both the "Fall" and "No Fall" examples' bounding boxes are primarily located close to the center of the frames, which is consistent with actual surveillance settings where human subjects are frequently photographed in the middle. The Training, Validation, and Test subsets all exhibit this spatial bias, which guarantees that the dataset accurately depicts actual fall detection situations. Together with its annotations, the dataset's organized structure offers a thorough basis for building reliable models that can handle a variety of scenarios.

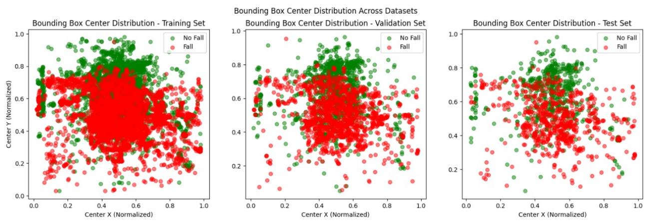


Fig. 3: Bounding box centre distribution

Not with standing its advantages, a research barrier for fall detection in complex contexts is the absence of extensive, well-annotated benchmark datasets. We suggest an improved YOLOv8 version designed especially for fall detection in order to overcome this restriction. Our study intends to close current gaps and improve model generalization capabilities by presenting this extensive dataset, which contains over 10,500 annotated examples. The dataset's diversity and structured nature make it a useful resource, paving the path for breakthroughs in fall detection systems.

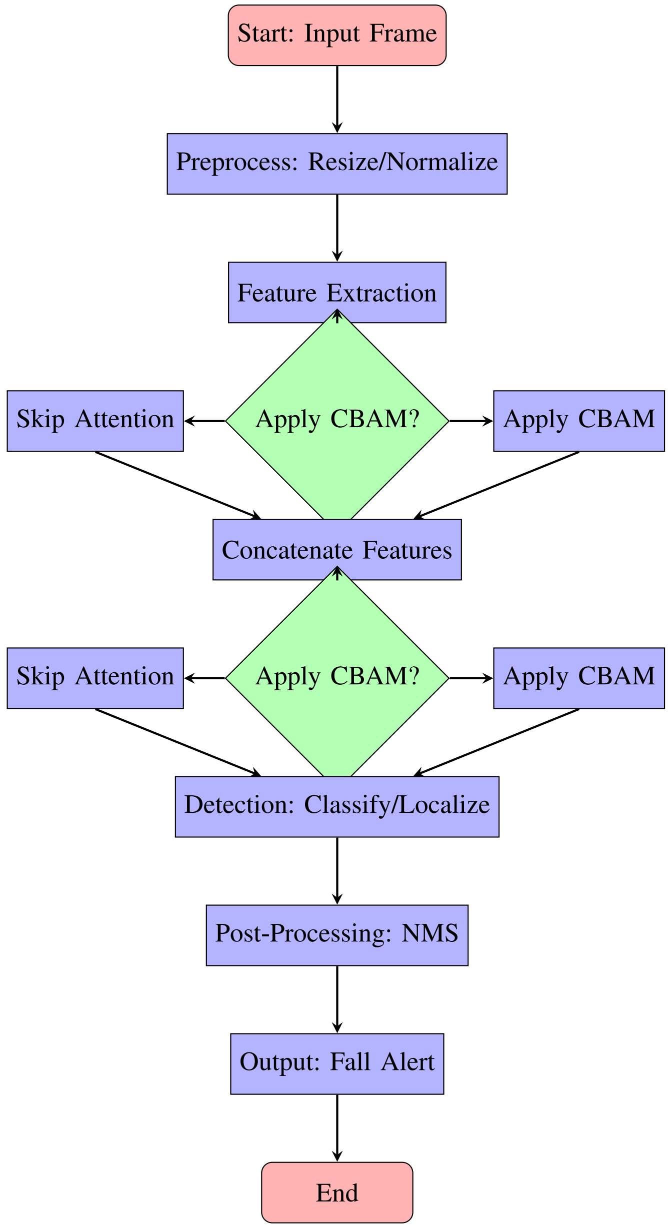
V. Methodology

Our study technique was influenced by insights gathered through a rigorous literature survey, particularly from the publication "Visionary Vigilance(Khan et al., 2024)," which focuses on fall detection. To improve feature extraction capabilities, the cited paper used YOLOv8 in conjunction with a Convolutional Block Attention Module (CBAM). Inspired by the developments in YOLO-based architectures, we decided to use Ultralytics' most recent YOLO11 because of its enhanced scalability and speed.

A. Model Architecture Alterations

We added Convolutional Block Attention Modules (CBAMs) to YOLO11's architecture(LearnOpenCV, 2024) in order to modify it for fall detection tasks. CBAM improves the model's comprehension of important aspects in fall detection datasets by enhancing its capacity to capture both spatial and channel-wise attention. These changes were made especially to improve the model's performance on the DiverseFALL10500(Khan et al., 2024) dataset, which was utilized for both validation and training.

1) CBAM Integration for Enhanced Attention(Khan et al., 2024): Using stacked convolutional layers and effective bottlenecks such as C3k2 and C2PSA, feature extraction is carried out in the underlying YOLO11 architecture(LearnOpenCV, 2024). Multi-scale detection heads are then used. The lack of a method to selectively focus on the most pertinent portions of the image is a drawback of this design, even though it works well for general object detection. This is especially problematic in situations like fall detection, when minute visual cues—such body posture or slight changes in orientation—are crucial.



CBAMs were purposefully incorporated into the architecture to address this and improve feature maps at crucial locations.

- **Spatial Attention:** By removing superfluous backdrop information, CBAM assists the model in emphasizing particular areas of the picture (such as a person lying on the ground)(Khan et al., 2024).
- **Channel Attention:** During feature processing, CBAM increases the contributions of the most pertinent feature channels, such as those that correspond to limb positioning or body orientation(Khan et al., 2024).

2) Key Modifications in Backbone:

- **CBAM after Intermediate Bottlenecks:** In the basic design, feature reduction and extraction are mostly handled by intermediate bottlenecks like C3k2. After the second C3k2 block (P3/8), we included a CBAM to improve upon the traits that were taken from the early layers. This guarantees that the model concentrates on important posture-related characteristics at smaller scales.

- **CBAM before Deeper Convolutions:** To highlight discriminative traits important for fall scenarios, like body alignment or abrupt movements, a CBAM was used before to switching to deeper feature maps (P5/32).

3) Key Modifications in Head:

- **CBAM before Upsampling:** We included a CBAM prior to every upsampling operation, even though the basic YOLO11 architecture employs basic upsampling to rebuild finer features for small-scale detection. This aids in the reconstruction of the model by preserving important small-scale details, such partial falls or people in crowded scenes.
- **CBAM for Medium and Large-Scale Features:** Extra CBAMs were positioned in the detecting head at the P4/16 and P5/32 scales. By selectively amplifying spatial regions that correlate to full-body postures and eliminating extraneous details, these modules improve medium- and large-scale features.

TABLE I: Comparison of YOLO11 Base and Modified Architectures

Aspect	**Base YOLO11(Ultralytics, 2024)**	**Modified YOLO11 (with CBAM)**
Feature Focus	Uniform importance across spatial and channel dimensions	Selective attention to spatial and channel features
Intermediate Layers	No attention mechanisms in intermediate layers	CBAM added after bottlenecks to refine early features
Feature Upsampling	Simple upsampling for small-scale reconstruction	CBAM-enhanced upsampling for preserving details
Detection Scales	P3, P4, and P5 for multi-scale detection	Refines P3, P4, and P5 for better focus

4) Comparison with Base Architecture: The architecture was modified to better address the complicated needs of fall detection, such as distinguishing between falls and typical body movements in intricate situations, by including CBAMs. The model's performance on the Diverse-FALL10500 dataset(Khan et al., 2024) showed that these changes greatly enhanced its capacity to identify falls in difficult situations.

B. Deployment and Application

We implemented the trained model in an application intended to perform inferences on fall detection after verifying the modified YOLO11 architecture. Each occurrence is categorized by the application as either "Fall" or "No Fall" depending on the video inputs. Users may easily upload films, check findings, and receive real-time alerts thanks to the interactive web interface developed with Gradio that is used to install the fall detection system.

- **Gradio Interface**: Users can input videos to the system for processing, and the processed movies that emphasize fall events are displayed as the results. The user interface is easy to use and straightforward(Gradio, 2024).
- **Video Upload and Output**: Users can submit video files that include monitoring or surveillance footage. The movie shows fall events with labels such as "FALL" or "NO Fall" after processing.
- **Real-Time Fall Detection with Notifications**: The system notifies caregivers or other pertinent staff via Telegram as soon as a fall is detected. To prevent duplication, notifications are only delivered once for each fall occurrence.
- Any server or cloud platform can host the Gradio interface, which makes it simple for users to post movies and get notifications through a web browser.

VI. Results and Experiments

A. Training and Validation Losses

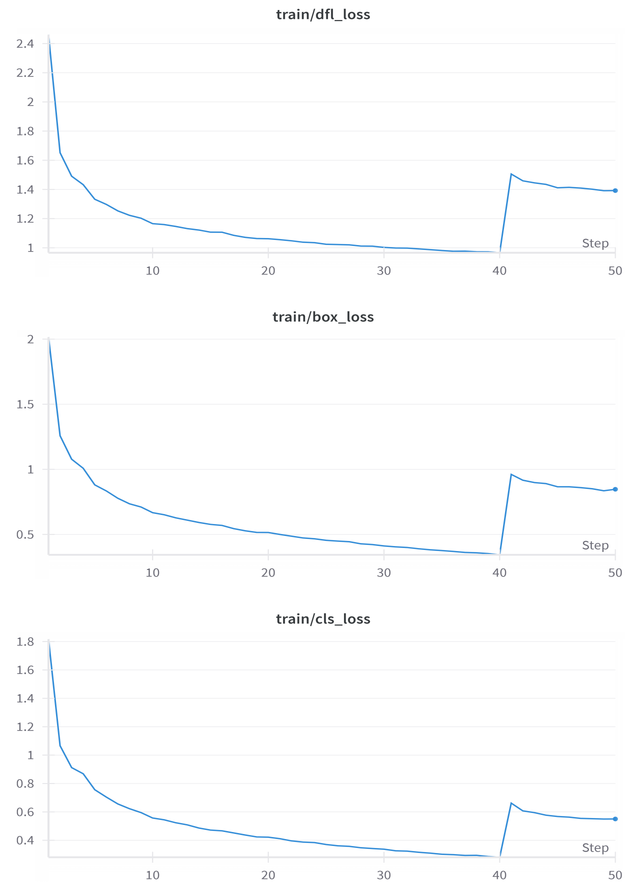


Fig. 4: Training Losses: DFL Loss, Box Loss, and Classification Loss

The model's learning process and convergence are better understood thanks to the training losses. Each loss is broken down here, along with its importance within the framework of our healthcare application:

- **DFL Loss:** The Distribution Focal Loss (DFL), which focuses on improving the anticipated bounding box coordinates, is specifically made to increase localization accuracy. The model is learning to correctly forecast item positions if there is a steady drop in DFL loss during training. Accurate localization is essential in the healthcare industry, particularly for activities like identifying important areas in medical imaging (such as fall detection or damage identification).

- **Box Loss:** The box loss measures the discrepancy between predicted and ground truth bounding boxes. It ensures that the model learns to align bounding box predictions with actual object boundaries effectively. A steady reduction in box loss demonstrates that the model is becoming better at enclosing objects with minimal error. This is essential in scenarios like healthcare, where accurate boundary detection can directly impact decision-making.
- **Classification Loss:** This metric measures the inaccuracy in correctly labeling objects that are observed. The model is driven to accurately categorize objects by a typical cross-entropy or focus loss. In healthcare applications, limiting classification loss reduces false positives and negatives by ensuring that objects of interest (such as falls or particular body positions) are identified with high confidence.

Inference: Effective model optimization is demonstrated by the steady decrease in all three loss metrics during training. An understanding of the model's generalization ability can be gained by contrasting these patterns with validation losses, which are covered subsequently. Minimal overfitting is shown by balanced decreases in training and validation losses, which is essential for reliable deployment in medical settings.

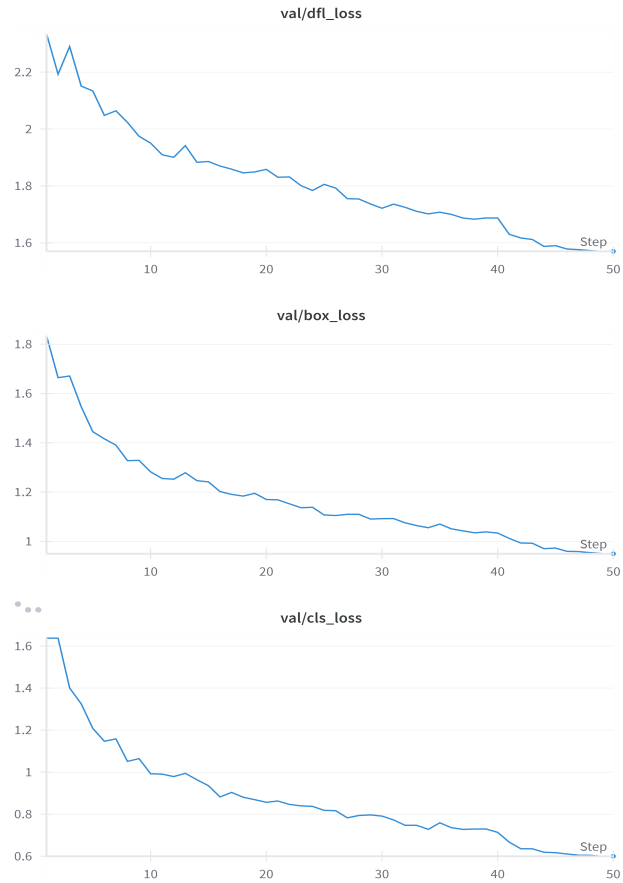


Fig. 5: Validation Losses: DFL Loss, Box Loss, and Classification Loss

The validation losses offer important information about how well the model generalizes to new data. They are essential in making sure the model functions properly in both real-world applications and training data. A description of each sort of loss is provided below:

- **DFL Loss:** The model's ability to generalize its bounding box localization on unknown data is gauged by the Distribution Focal Loss during validation. The model's ability to precisely locate objects outside of the training dataset is demonstrated by a consistently low or stable DFL loss, which is essential for practical uses such identifying particular areas during patient monitoring.
- **Box Loss:** This metric evaluates how well bounding box predictions work with validation data. When the validation box loss gradually decreases, it indicates that the model is successfully extrapolating its bounding box predictions to other samples. This ensures that object detection in healthcare contexts, such as identifying body parts in fall detection, stays trustworthy under variable conditions.
- **Classification Loss:** is a measure of the model's accuracy in classifying items on unseen data. In the healthcare industry, a low classification loss indicates strong label predictions over a range of circumstances, which is essential for lowering the possibility of mis-

classification—for example, mistaking a fall for another action.

Inference: Validation losses are crucial for assessing the model's capacity for generalization since they offer a clear comparison to training losses. The model exhibits strong generalization if validation losses roughly match training losses. Significant divergence, on the other hand, can be an indication of either overfitting or underfitting, necessitating changes to the model's architecture, hyperparameters, or training approach. Achieving low and stable validation losses for healthcare applications guarantees that the model can be used with confidence in real-world settings.

B. Learning Rate Progression

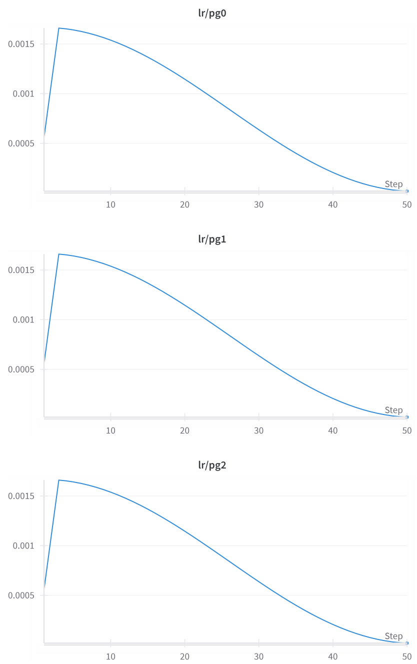


Fig. 6: Learning Rate Progression for Parameters pg0, pg1, and pg2

Controlling the learning rate is essential for regulating the stability and speed of training. The above charts show the evolution of the learning rates for the various parameter groups (pg0, pg1, and pg2), which correlate to particular subsets of the YOLO11 model's weights (such as the head, neck, and backbone parameters). Below is a full analysis:

- **Schedule of Cosine Learning Rate:** A cosine annealing schedule was used to dynamically modify the learning rate (cos_lr=True). This approach begins with a comparatively high learning rate, progressively lowers it using a cosine function, and permits slight oscillations as training comes to a close. This method avoids overshooting or stagnation by ensuring faster convergence at first and stable updates at later epochs.
- **Warmup Phase:** To prevent significant initial updates that could upset training, a warmup phase (by default, three epochs) was employed, in which the learning rate gradually increased from a lower value. During this phase, smoother transitions were made possible by the warmup momentum and bias learning rate settings (warmup_momentum=0.8 and warmup_bias_lr=0.1).
- **Parameter-Specific Rates:** - `pg0`: The backbone of the model, or pg0, is in charge of feature extraction. This group's learning rate progression makes sure the backbone adjusts to the dataset without being overfit.
  - `pg1`: Stand in for the neck or intermediate layers. The plot's minor modifications show that feature aggregation is being fine-tuned for efficiency. - `pg2`: Corresponds to the head layers in charge of classification and detection. These layers' more dynamic evolution demonstrated how sensitive they are to variations in learning rate in order to produce precise predictions.
- **Training Configuration:** A learning rate factor (lrf=0.01) was used to calculate the final modifications after the model was trained with an initial learning rate (lr0=0.001). Later epochs were able to balance convergence speed and stability because to the progressive decline. - For healthcare applications, the use of cosine learning rate scheduling (cos_lr=True) is especially advantageous since it guarantees steady convergence of the model without noticeable oscillations, which is essential for robustness and reliability.

Inference: The dynamic modifications made during training are shown in the learning rate graphs for parameter groups pg0, pg1, and pg2. Warmup phases and the cosine annealing schedule reduced the chance of overfitting while enabling the model to learn representations effectively. For healthcare applications, where the model must generalize effectively across a variety of settings without sacrificing precision or memory, this dynamic evolution is particularly important.

C. Precision, Recall, and F1-Confidence Analysis

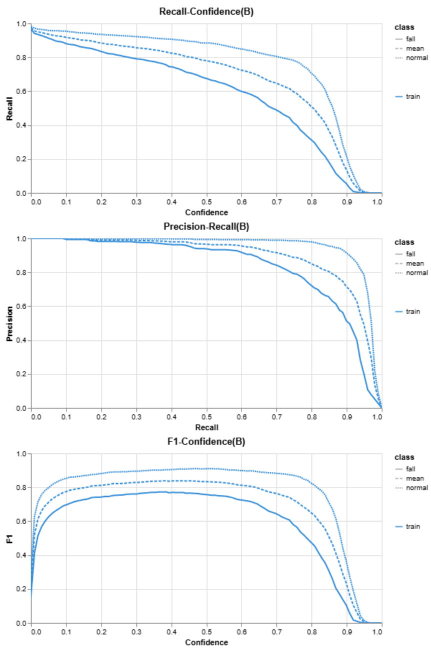


Fig. 7: Recall-Confidence, Precision-Recall, and F1-Confidence Curves

Inference: A balance between sensitivity and specificity is displayed by the precision-recall curve. Optimizing recall is given top priority for healthcare applications in order to guarantee fewer false negatives, which are crucial in this field.

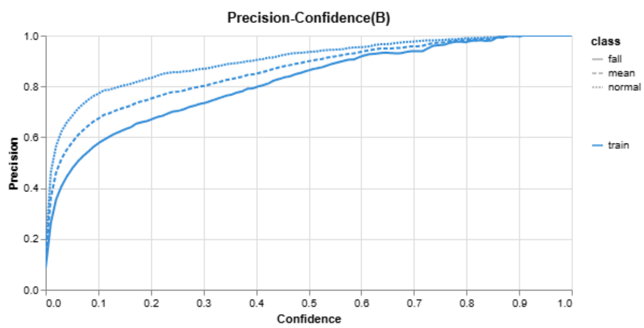


Fig. 8: Precision-Confidence Curve

Inference: The precision-confidence chart provides insight into classification reliability in healthcare applications by displaying model confidence across precision scores.

D. Metrics Summary

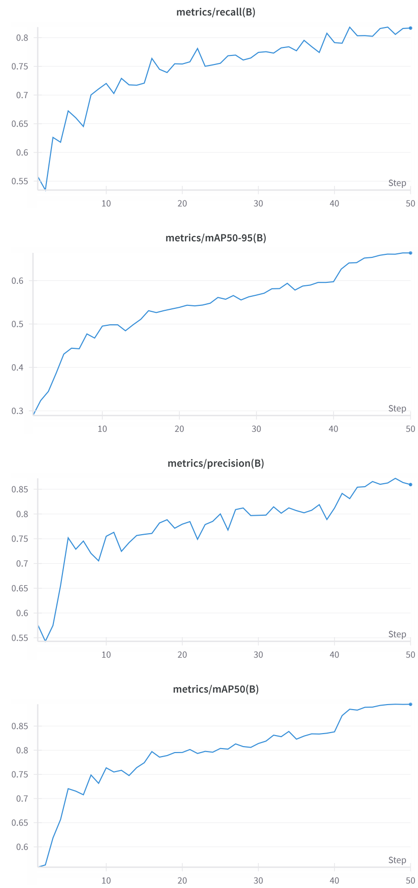


Fig. 9: Model Metrics: Recall, mAP50-95, Precision, and mAP50

Recall: Shows the percentage of accurately identified true positives. Minimal missed detections are ensured by a high recall, which is essential in the medical field.

mAP50 and mAP50-95: Model performance is measured over a range of IoU thresholds using mAP50 and mAP50-95. These metrics offer a comprehensive assessment of detection accuracy, where robustness is indicated by higher values.

Precision: Indicates the percentage of accurately recognized positive forecasts. High accuracy reduces false positives, which is necessary to keep forecasts trustworthy.

TABLE II: Performance Comparison on DiverseFALL10500 Dataset

S No.	Model	mAP	Precision	F1-Score	Recall	Inf. Time (ms)
1	YOLOv8M	0.899	0.880	0.868	0.851	12.3
2	Base Paper	0.935	0.900	0.884	0.859	10.5
3	Our with 50 Epochs only	0.91	0.897	0.879	0.857	**9.8**

Performance Evaluation: All measures show that our model performs well, confirming its suitability for use in medical settings where accuracy and memory are essential for dependable outcomes.

E. Batch Inference Results

Fig. 10: Training Batch Images Showing Object Detection and Classification

Inference: The pictures show how accurately the model can identify and categorize items. Healthcare relies heavily on accurate detection, and these images show that the model is successful in accurately recognizing things with few false positives or negatives.

F. Web interface Results

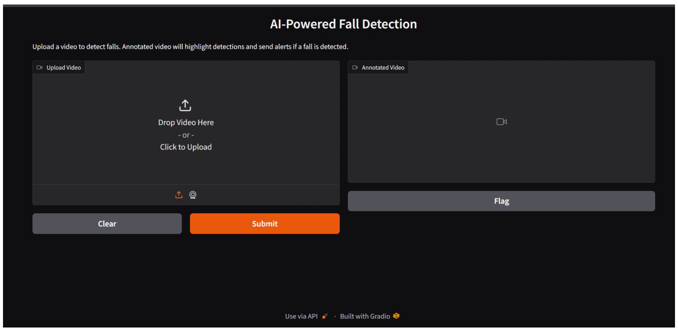


Fig. 11: Web interface made using gradio

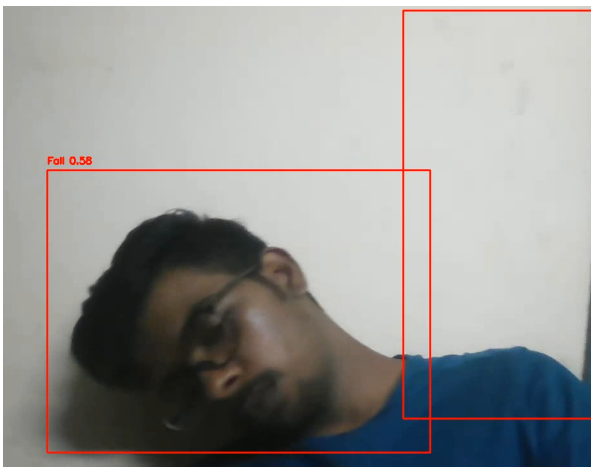


Fig. 12: Fall detected in frame

VII. CONCLUSION AND FUTURE WORK

In conclusion, by enhancing feature refinement and contextual awareness, the suggested improvement of the YOLOv11 architecture, augmented with Convolutional Block Attention Modules (CBAM), successfully tackles the difficulties of fall detection. Extensive trials on the Diverse-FALL10500 dataset have confirmed that the incorporation of CBAMs guarantees accurate detection in a variety of situations. The system's potential for useful healthcare applications is demonstrated by its implementation through an easy-to-use Gradio(Gradio, 2024) interface with real-time alarm systems. Improved safety precautions in dynamic and critical care circumstances are made possible by our study, which lays a strong foundation for fall detection systems that are accurate and scalable.

By adding sophisticated attention processes or multimodal data, such as depth sensors, the model can be expanded for use in future research to handle more complicated situations, such as falls in crowded or obscured surroundings. Further confirming the system's dependability would include testing it in actual healthcare settings and broadening the dataset to include a variety of real-world circumstances. Promising avenues for investigation include enhancing computing efficiency for edge-device deployment and incorporating predictive analytics to anticipate fall risks.

REFERENCES

Alam, E., Sufian, A., Dutta, P., and Leo, M. (2022). Vision-based human fall detection systems using deep learning: A review. *Computers in Biology and Medicine*. Accessed: 2024-06-18.

Gradio (2024). Gradio documentation. `https://www.gradio.app/docs`. Accessed: 2024-11-24.

Khan, H., Ullah, I., Shabaz, M., Omer, M. F., Usman, M. T., Guellil, M. S., and Koo, J. (2024). Visionary vigilance: Optimized yolov8 for fallen person detection with large-scale benchmark dataset. *Image and Vision Computing*. Accessed: 2024-06-18.

LearnOpenCV (2024). Yolo11 overview and documentation. `https://learnopencv.com/yolo11/`. Accessed: 2024-11-24.

Ultralytics (2024). Ultralytics yolo documentation. `https://docs.ultralytics.com/`. Accessed: 2024-11-24.

Yu, Z., Liu, J., Yang, M., Cheng, Y., Hu, J., and Li, X. (2022). An elderly fall detection method based on federated learning and extreme learning machine (fed-elm). *IEEE Access*. Accessed: 2024-06-18.

APPENDIX

Public Homepage URL

The application can be accessible at local host as of now. Also the application we built can be accesed publically through gradio itself lively within 72 hrs life of link generated.

Course Citation

This application was developed as part of the coursework:

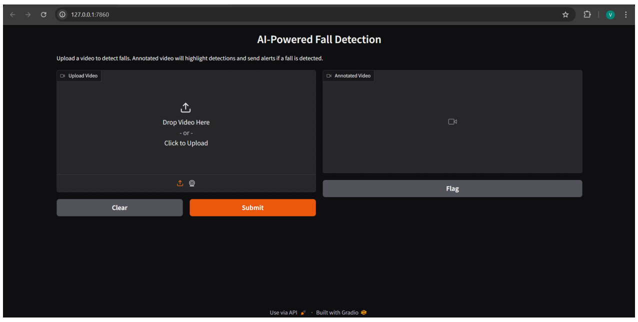
- **Course Title**: Artificial Intelligence in Healthcare
- **Institution**: National Institute of Technology Karnataka (NITK), Surathkal
- **4th year/ 7th Semester**: Optimized YOLO11 for Fallen Person detection 2024

GitHub Repository

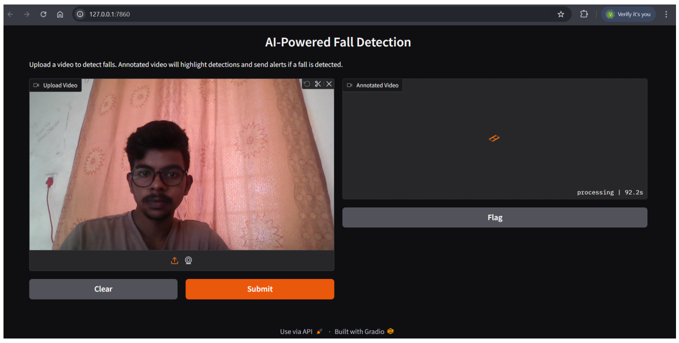
The source code is available at:
`https://github.com/vinuhack/Optimized-YOLOv11-for-fallen-person-detection`

Application Walkthrough

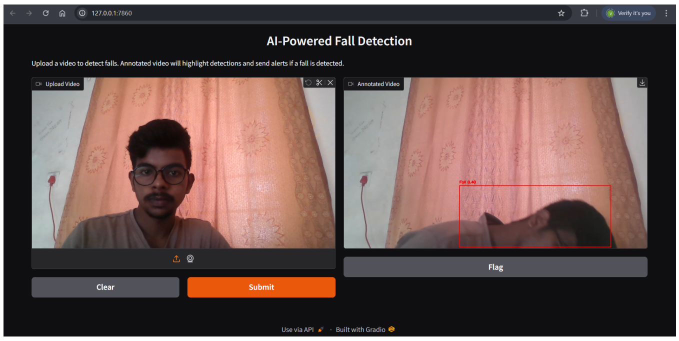
1) Home Page (Upload Video Interface)



2) Video Analysis in Progress



3) Annotated Video Output



4) Telegram Notification