REVIEW ARTICLE

Visual Intelligence for Automated Fall Sensing: A Systematic Review of Architectures, Datasets, and Evaluation Gaps

Babar Zeb[1,†], **Muhammad Talha Usman**[2,†], **Habib Khan**[2,†,*], **Alexandros Gazis**[3,†], **Stylianos Pappas**[4,†], **Muhammad Faizan Omer**[5,†] and **Nasir Rahim**[2,†,*]

[1] Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, NUST, Islamabad, Pakistan
[2] School of Computing, Gachon University, Seongnam-si 13120, Republic of Korea
[3] Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67100, Greece
[4] Electrical Engineering and Computer Science, Hellenic Naval Academy, Terma Chatzikyriakou, Piraeus 18539, Greece
[5] Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology (NUST), Rawalpindi, Pakistan

Abstract

Falls are a major cause of injury, hospitalization, and loss of independence among older adults, spurring interest in visual intelligence-based automated fall detection for timely response and continuous monitoring. This article presents a systematic review of such systems, focusing on YOLO-based approaches. Following PRISMA guidelines, the review covers 2016–2025 literature, identifying 637 records and including 63 studies after screening. We examine datasets, preprocessing strategies, evaluation protocols, metrics, and hardware platforms, comparing reported accuracy, efficiency, and real-time feasibility across different designs. Evidence is strongest for YOLOv3 through YOLOv9, while evidence for YOLOv1–YOLOv2 and YOLOv10–YOLOv12 remains limited or preliminary. Reviewed studies show steady performance gains on public datasets, but key limitations persist: most evaluations use staged or lab-controlled data, cross-dataset and cross-site testing are uncommon, and deployment factors like end-to-end latency, edge device performance, and long-term stability are under-reported. This review consolidates current practices and identifies priorities for more standardized, deployment-relevant reporting.

Keywords: fall detection, visual monitoring, object detection, systematic review, edge devices, real-time, assisted living.

1 Introduction

As the global population ages, maintaining older adults' safety and independence is important. Falls

Submitted: 16 November 2024
Accepted: 20 May 2026
Published: 27 June 2026

Vol. 3, **No.** 2, 2026.
10.62762/TSCC.2026.604481

***Corresponding authors:**
✉ Habib Khan
habibkhan@ieee.org
✉ Nasir Rahim
nrahim3797@ieee.org

† These authors contributed equally to this work

Citation
Zeb, B., Usman, M. T., Khan, H., Gazis, A., Pappas, S., Omer, M. F., & Rahim, N. (2026). Visual Intelligence for Automated Fall Sensing: A Systematic Review of Architectures, Datasets, and Evaluation Gaps. *ICCK Transactions on Sensing, Communication, and Control*, 3(2), 90–108.

remain a leading cause of injury for this group and often result in hospitalization, long-term disability, or death. Health systems require usable and affordable solutions that support aging in place and reduce avoidable admissions. These solutions must detect rare events with high sensitivity. They must keep stable precision to limit alarm fatigue. They must maintain low latency to ensure timely responses in real-world settings. Researchers in medicine and computer vision focus on fall detection that is accurate and real-time, with clear operating points and deployment constraints. The public health burden is substantial across regions and income levels. According to a 2021 WHO report, approximately 37.3 million falls each year require medical attention, with approximately 684,000 deaths annually. Falls rank as the second leading cause of unintentional injury mortality [1]. As the number of older adults increases, the risk of fall-related injury also increases, especially for those aged 65 years and above. Outcomes include fractures, head trauma, functional decline, fear of falling, and loss of independence. These outcomes drive caregiver workload, length of stay, readmissions, and community support needs. Practical systems should integrate with existing workflows in homes and facilities. They should work with clinical and social care services. They should provide actionable alerts for review. Privacy protection and informed consent are necessary for adoption and regulation. These requirements shape choices in sensing modality, data management, model design, and evaluation. Early foundational work established the core principles and detection methods that underpin modern fall detection system design [2]. They also motivate standardized reporting and practical Internet of Things (IoT) deployment, ensuring that results are comparable and transferable across settings [3].

Traditional fall detection relies on wearable devices or environmental sensors [4, 5]. These approaches exhibit poor compliance, placement variability, limited coverage, and high deployment costs [4, 5]. Users may forget to wear devices or charge them. They may also wear them in ways that reduce signal quality. Wrist or hip placement alters the signal, and daily activities can resemble falls. Environmental sensors often need room-by-room installation. They also need calibration and ongoing maintenance. Vision-based systems use cameras and deep learning to detect falls without worn devices [6–8]. One-stage detectors, such as YOLO, are attractive for real-time monitoring when the model and hardware match the target setting. Recent work has also explored efficient attention modules within YOLO-based human fall detection for real-time one-stage detection [9]. These systems can fit indoor settings such as hospitals, care homes, and smart homes. They may use a single camera to enable efficient operation on low-power hardware, reducing the need for multiple cameras. They can use red, green, blue (RGB), depth, or thermal streams when available. Advances in edge computing and embedded artificial intelligence (AI) support efficient operation on low-power hardware, which reduces installation and maintenance costs while keeping data local [3, 27]. Model compression and optimized inference can further reduce latency and memory use on resource-limited devices [27, 33].

1.1 Limitations in Existing Research

YOLO-based fall detection has improved, yet significant challenges remain. Performance degrades under low illumination [10], occlusions and partial views, and cluttered backgrounds [12]. Occlusions include self-occlusion, inter-person occlusion, and partial views due to furniture or tight spaces [12]. Camera viewpoint, height, and lens field of view influence posture cues near the floor and change apparent motion during collapse [12]. Many models are trained on simulated or staged data and exhibit reduced performance in real environments [13]. Falls are rare events, which leads to class imbalance and fewer positive samples for learning. This imbalance requires careful threshold selection, cost-sensitive training, and reporting beyond accuracy alone. Although YOLO targets real-time operation, throughput on resource-constrained devices is not always sufficient for home or care settings. End-to-end latency, sustained frame rate, thermal limits, and memory pressure affect stability during continuous monitoring. Prior work has explored data augmentation, ensemble methods, and transfer learning, but protocols vary, complicating comparisons across studies. Split strategies sometimes leak subjects or scenes between training and testing sets, inflating results and masking generalization limits. Few studies evaluate on out-of-distribution data or adapt models to new deployment sites [13]. Site-specific background, lighting profiles, and camera geometry shifts can alter data distributions and reduce recall if not addressed. Recent fall-detection research has highlighted the impact of occlusion, viewpoint variation, and real-world scene complexity on performance, motivating explicit robustness analysis for deployment-oriented evaluation [11]. These

Table 1. Summary of YOLO variants and their applications in fall detection.

Variant	Year	Dataset(s)	Architectural Improvements	Relevance to Fall Detection	Reference
YOLOv1	2016	URFD	Unified CNN, regression-based detection	Real-time; weak localization for small objects	[24]
YOLOv2	2017	URFD	Anchor boxes, batch norm, multi-scale	Higher speed and precision	[26, 27]
YOLOv3	2018	Le2i (applied studies)	Darknet-53, multi-scale features (pyramid-style)	Better handling of small targets; early edge use	[17, 25]
YOLOv4	2020	NTU RGB+D	CSPDarknet53, SPP, PANet	Higher accuracy on multi-pose falls	[18, 30, 32]
YOLOv5	2021	CAUCAFall, URFD, NTU RGB+D	PyTorch impl., AutoAugment, practical model scales	High precision; real-time setups	[19, 20, 33, 34]
YOLOv6	2022	N/A	Anchor-free head, efficient conv, label assignment	Low-latency edge potential	[21, 35, 36]
YOLOv7	2022	DiverseFall	E-ELAN, RepConv, optimized heads	Robust under occlusion in applied reports	[37–39]
YOLOv8	2023	NTU RGB+D, DiverseFall	Decoupled head; applied backbones and data diversity	Better posture sensitivity in applied studies	[31, 40, 41, 50]
YOLOv9	2024	Synthetic; multi-visual fall sets	Programmable Gradient Information (PGI)	Strong accuracy-efficiency trade-offs; limited fall evidence	[14, 15]
YOLOv10	2025	COCO family; no public fall dataset reported	Deployment focus; related work on efficient attention and token reduction	Real-time edge readiness (preliminary for falls)	[42, 45, 48]
YOLOv11	2025	Le2i; URFD; MCFD (applied studies)	Multi-query sparse attention, tracking and pose cues	Applied reports on fall datasets	[46, 47]
YOLOv12	2025	COCO family; no public fall dataset reported	Transformer-style detector direction; sparse attention approximations	Preprint-stage; no fall-data results reported	[43, 44, 52]

Note: Architectural components referenced in the table draw on high-impact venues: FPN [17] and PANet [18] (CVPR), AutoAugment [19] and EfficientDet [20] (CVPR), RepVGG [37] and ConvNeXt [31] (CVPR), FlashAttention [42], Linformer [43], and Performer [44] (NeurIPS).

gaps warrant further investigation with standardized benchmarks, fixed thresholds, paired GPU and edge reports, and precise temporal evaluation. Recent efforts include multimodal or multi-sensor fall-detection and staged-to-wild benchmarks [13, 14], as well as newer detector backbones that may improve feature quality [15]. Adoption and reporting remain inconsistent across studies, underscoring the need for consolidated reviews, metrics (e.g., precision, recall, F1-score, and mean average precision (mAP)), and transparent protocols that reflect deployment constraints and privacy requirements.

These gaps directly motivate the research questions addressed in this review. RQ1 examines how YOLO architectures evolved to address challenges such as occlusion, partial views, clutter, and low illumination, and is answered in Subsection 4.1 (YOLO Variants and Architectural Evolution). RQ2 addresses variation in datasets, preprocessing practices, and evaluation protocols, and is answered in Subsection 4.2 (Datasets Used in YOLO-based Fall Detection) together with the evaluation metrics discussion in Section 4.3. RQ3 focuses on comparative performance under environmental and hardware constraints, and is answered in Subsection 4.4 (Performance Comparison of YOLO Variants). RQ4 synthesizes the remaining challenges, limitations, and future directions that shape YOLO-based fall detection research, and is addressed in Section 5 (Conclusion and Future Work).

1.2 Key Contributions

This article traces the evolution of YOLO-based fall detection from YOLOv1 to YOLOv12. It summarizes the literature and identifies remaining gaps. This section covers the following points.

- We examined research papers that applied various versions of YOLO to fall detection.

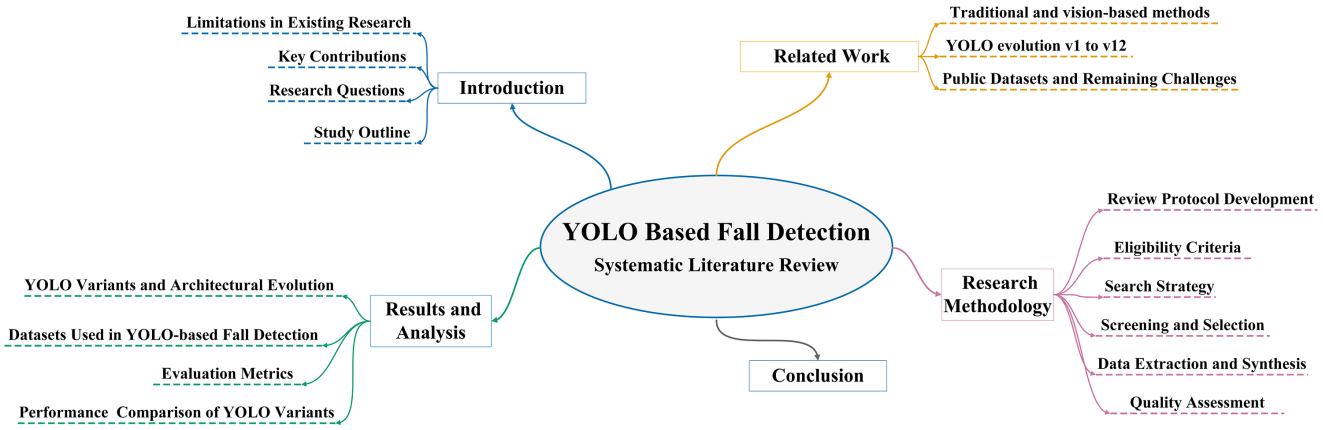

Figure 1. Paper organization and study outline for this systematic review of YOLO-based fall detection.

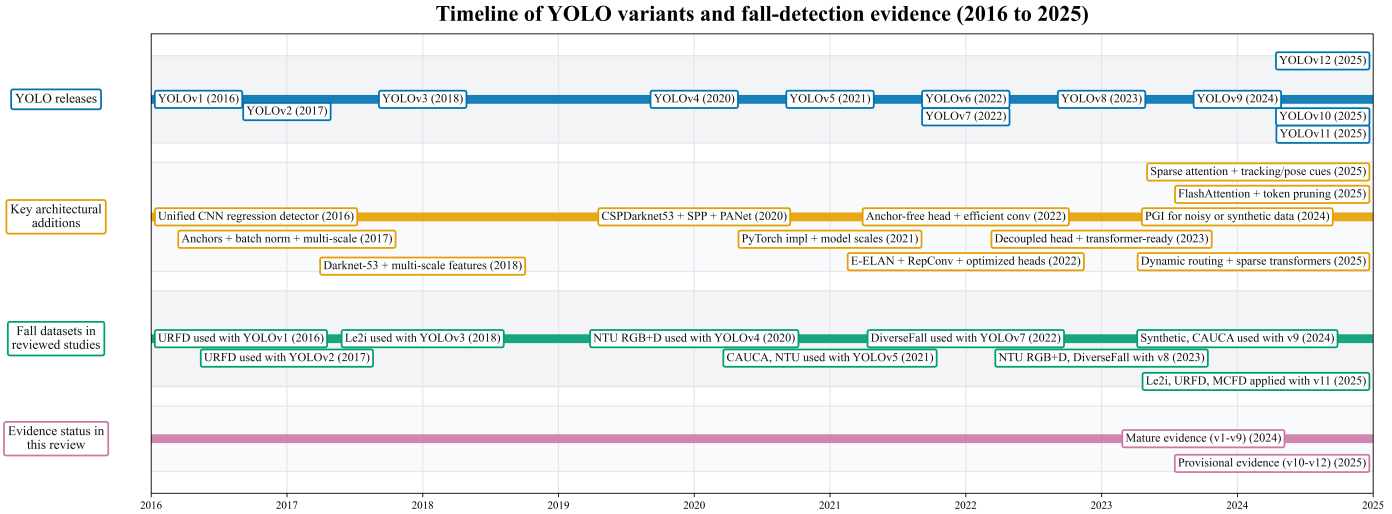

Figure 2. Timeline of YOLO variants and fall detection evidence. The figure aligns YOLO releases with major architectural additions, representative dataset-version pairings reported in the reviewed literature, and the evidence-status summary used in this article.

- We organize the datasets, YOLO variants, evaluation metrics, and reported results used in these studies.
- We compare YOLO versions across datasets to study trade-offs among accuracy, speed, generalization, and compute cost.
- We identify open problems, including multimodal fusion, privacy constraints, and robust edge deployment on resource-limited devices.

1.3 Research Questions

This article answers the following research questions:

- **RQ1:** How have YOLO architectures evolved from v1 to v12 in the context of fall detection?
- **RQ2:** What datasets, preprocessing steps, and evaluation metrics are most common in YOLO-based fall detection?
- **RQ3:** Which YOLO variants perform best under specific environmental or hardware constraints?
- **RQ4:** What challenges, limitations, and future directions shape YOLO-based fall detection research?

1.4 Study Outline

This article follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines for study selection and literature analysis. The paper is organized as follows: Section 1 introduces the topic and background. Section 2 reviews related work. Section 3 describes the research methodology. Section 4 reports results and analyzes the findings. Section 5 concludes the paper and outlines directions for future work. Figure 1 summarizes the paper structure and the key topics in each section.

2 Related Work

Fall detection has progressed from rule-based sensing to learned, vision-first systems. Early efforts used simple thresholds on wearable signals, which worked in lab settings but failed in homes with varied routines [4]. Figure 2 summarizes the YOLO release timeline and key architectural additions across versions. It also maps representative dataset-version pairings from the reviewed literature, allowing readers to trace approximate adoption timing and the lag between YOLO releases and their use in fall detection. This article identifies where evidence is mature (v1-v9) and where it remains limited (v10-v12). As cameras and edge hardware improved, single-shot detectors made real-time analysis practical in rooms and corridors. YOLO is key to this shift because it enables high frame rates while maintaining stable localization at common indoor scales. We examine how YOLO evolved from v1 to v12 and explain what each release adds for posture change, small bodies near the floor, and crowded scenes. We relate these changes to the features in Table 1 so that readers can map the year, backbone, detection head, and training choices to fall detection requirements. Table 1 is intended as a qualitative summary of architectural evolution and its relevance to fall detection, rather than a standardized performance ranking, because the reviewed studies differ substantially in datasets, data splitting strategies, evaluation thresholds, IoU settings, and reporting protocols. We also note design ideas that influence YOLO training and use, such as residual connections, pyramid features, path aggregation, learned augmentation, and efficient attention [16–19]. These ideas aim to improve small-object detection and reduce false alarms in clutter. They also aim to reduce compute and memory requirements for edge deployment. We link them to deployment needs in care settings, including stable recall at fixed latency, predictable behavior under occlusion, and simple deployment on constrained hardware. Finally, we connect detector updates to neck and assignment changes that improve robustness in indoor scenes [20, 21]. Together, these advances explain why recent YOLO versions fit room-scale monitoring and real-time use.

2.1 Traditional and Vision-Based Approaches to Fall Detection

Sensor-based methods measure motion directly. Common wearable units include accelerometers and gyroscopes on the wrist, hip, or chest. Ambient sensors include pressure mats and doorway beams placed in specific rooms. These pipelines are simple to deploy for a single user. They tolerate poor lighting and can run for days on a small battery. They also face practical limits. People forget to wear devices, charge them late, or place them in ways that distort the signal [4, 5]. Fixed sensors cover only the instrumented area, so blind spots remain in large homes and wards. Threshold rules are triggered by rapid motion, which confuses falls with vigorous activity unless models incorporate context [4]. Long-term adherence and maintenance drive cost and missing data [4, 5]. Comparative analysis of accelerometer-based fall detection algorithms demonstrates strong performance in controlled trials but reduced reliability when activity diversity and irregular movement patterns vary across individuals [23]. These limits motivate complementary sensing to detect posture, contact with the floor, and scene layout.

Vision-based methods observe posture and scene context with red, green, and blue (RGB) or depth cameras. Early approaches extracted silhouettes and motion descriptors and then classified events using Support Vector Machines (SVMs), Hidden Markov Models (HMMs), or k-Nearest Neighbors (k-NN) [6, 8]. These pipelines operated in controlled rooms but dropped frames or misread poses when lighting changed, people overlapped, or the viewpoint changed [6]. Background subtraction and contour features were sensitive to shadows and camera jitter. Deep detectors improved stability by learning features directly from images. YOLO detects people in a single pass, supports multi-scale heads for small targets near the floor, and maintains low latency sufficient for timely alerts on commodity GPUs [24, 25]. Recent fall-detection studies also employ YOLO variants to detect falls or fallen individuals in indoor scenes [22]. Multi-camera layouts and depth streams reduce occlusion and improve view coverage [8]. Edge deployment keeps frames local and reduces bandwidth use, thereby enhancing privacy and reducing costs. Open issues remain. Models face class imbalance because actual falls are rare, there are gaps in generalization across homes and cameras, and they are sensitive to viewpoint and floor clutter [6]. Robust practice includes subject-wise splits, fixed intersection over union (IoU) thresholds, and transparent reporting of precision, recall, F1, mean average precision (mAP), and frames per second (FPS) [6]. Many teams now combine inertial signals with video to fuse body dynamics with scene cues, which raises recall while limiting false alarms [6].

2.2 YOLOv1 to YOLOv3: Foundational Models for Real-Time Detection

Convolutional Neural Networks (CNNs) enabled direct learning from images, and residual learning improved the training of deeper networks [16]. YOLOv1 introduced a unified, single-pass detector for real-time use [24]. Table 1 lists its year, a typical dataset, and its regression-based design. YOLOv1 was fast, but it struggled with small bodies and partial views. YOLOv2 added anchor boxes, batch normalization, and higher input resolution [26]. This corresponds to the "Architectural Improvements" entry for v2 in the table. In UR Fall (URFD), studies using v2-style updates report higher precision and recall than v1 baselines, although these gains are reported under study-specific conditions rather than comparisons [27]. YOLOv3 adopted the deeper Darknet-53 backbone and multi-scale prediction [25], which is consistent with pyramid features known to support small-object detection [17]. In fall detection, this is relevant to small or partially visible persons, but the reviewed literature does not provide matched ablation studies that isolate the effect of each architectural change under common settings. Separate fall pipelines also model body posture and spatio-temporal evolution, which motivates posture-aware cues that can complement detection stages [28]. Overall, v1 to v3 established single-shot speed, anchors, and multi-scale features, and their relevance to fall detection should be interpreted as an evidence-informed qualitative trend rather than a standardized quantitative progression.

2.3 YOLOv4 to YOLOv5: Balancing Accuracy and Efficiency

YOLOv4 strengthened feature reuse and neck design. It used CSPDarknet53, Spatial Pyramid Pooling (SPP), and Path Aggregation Network (PANet) [30]. These design choices subsequently influenced practical fall detection implementations, where improved YOLOv5s-based architectures have been applied to elderly fall detection with enhanced backbone and neck configurations [62]. PANet improves top-down and bottom-up fusion [18]. CSP ideas reduce computational cost while maintaining accuracy [31]. These map to the "Architectural Improvements" cell for v4. On NTU RGB+D, v4 improved multi-pose fall detection [32]. YOLOv5 brought a PyTorch codebase, strong augmentations, and practical model sizes [33]. AutoAugment supports data diversity [19]. Efficient multi-scale fusion inspired by BiFPN is common in modern necks [20]. The table lists CAUCAFall and NTU as example datasets for v5. On CAUCAFall, v5s reported high precision for posture states [34]. Variants such as YOLOv5n and YOLOv5x, along with subsequent releases including YOLOv8 and YOLOv10, demonstrate how model scaling enables users to select operating points that balance speed and accuracy for different deployment contexts [29]. This fits edge needs, and the "Relevance" column notes real-time setups. Practice also improved with better label assignment and thresholds. Works such as ATSS and OTA have shaped assignment rules that many detectors adopt [21, 35]. These advances improve stability in cluttered rooms and low-light conditions.

2.4 YOLOv6 to YOLOv12: Transformer Integration and Edge Readiness

Later versions focus on deployment and efficient attention. YOLOv6 adopts efficient backbones and an anchor-free head for industrial settings [36]. Re-parameterized blocks speed inference after training [37]. YOLOv7 refines training and scaling and reports strong results on the Microsoft Common Objects in Context (MS COCO) benchmark [38]. Applied fall detection papers use newer YOLO backbones and pose cues, but protocols vary across datasets [39–41]. YOLOv9 introduces Programmable Gradient Information (PGI) and a new backbone design to improve the accuracy-efficiency trade-off [15]. YOLOv11 appears in applied fall detection work evaluated on Le2i and on URFD and MCFD [46, 47]. Evaluation is also expanded to multi-modal fall datasets such as MUVIM [14]. Optimizer selection has also been shown to affect training convergence and detection accuracy across YOLOv8 and YOLOv11 variants, with comparative analysis revealing meaningful differences in small object detection performance [51]. Recent releases focus on faster attention and token efficiency. FlashAttention reduces memory traffic and improves speed [42]. Linformer and Performer approximate attention with linear complexity [43, 44]. Token merging reduces the number of tokens with minimal loss in quality [45]. These ideas motivate the "Architectural Improvements" entries for v10 to v12 in Table 1. YOLOv10 targets end-to-end deployment with NMS-free design [48]. YOLOv12 is proposed as an attention-centric real-time detector, and fall benchmarks remain sparse in current reports [52]. The broader landscape of real-time Transformer-based detectors, including RT-DETR [49], provides architectural context for understanding the direction of attention-centric designs that inform YOLOv12's development. In Table 1, the "Relevance"

Table 2. Search results from selected databases.

Search Term	IEEE	Springer	Elsevier	ACM	T&F	MDPI	Scopus
YOLO and Fall Detection	36	29	17	20	34	13	23
Vision-based Fall Detection and Deep Learning	29	18	17	3	23	10	10
Real-time Fall Detection and YOLO	22	15	8	3	11	8	7
YOLO and Elderly Monitoring	24	18	17	20	23	11	13
Deep Learning and Fall Detection and YOLO	29	26	17	19	34	15	15

Table 3. Elements of data extraction and synthesis.

Sr. #	Description	Details
1	Bibliography	Author names, publication year, publication venue
2	Databases	IEEE Xplore, SpringerLink, Elsevier ScienceDirect, ACM Digital Library, Taylor & Francis
3	Model Architecture	YOLO variant used (e.g., YOLOv3, YOLOv4, YOLOv5, YOLOv9)
4	Dataset Used	URFD, SDUFall, Le2i, DiverseFall, CAUCAFall, custom datasets
5	Evaluation Metrics	Accuracy, precision, recall, F1-score, IoU
6	Deployment Details	Real-time use, reported FPS or latency, edge computing, and hardware needs
7	Limitations	Dataset imbalance, lighting, occlusion, generalization issues
8	Methods	Fall detection method (single-stage or multi-stage)
9	Technologies	Frameworks and tools (PyTorch, TensorFlow, OpenCV)
10	Application Context	Elderly monitoring, healthcare, surveillance

column marks v10 to v12 as provisional in this article.

2.5 Public Datasets and Remaining Challenges

Public datasets support comparison and progress. UR Fall (URFD) offers controlled scenes with RGB-D. It includes clear fall events but limited variation. Le2i uses home-like rooms with more clutter and fewer fall samples. NTU RGB+D provides RGB, depth, infrared, and skeleton streams, which support temporal reasoning and multimodal fusion. CAUCAFall includes elderly subjects and realistic room layouts. It is realistic but not fully public. DiverseFall encompasses diverse indoor settings, camera heights and angles, and a broader range of subjects. It also provides YOLO-format annotations, which support consistent training and evaluation, although some classes still have modest sample counts [50]. Synthetic sets scale easily and allow control of pose, lighting, and camera, but they often need adaptation to real homes and wards. Challenges persist across sets. Illumination changes reduce recall. Background clutter, camera placement, and viewpoint shifts affect detection. Class imbalance is common because falls are rare. Domain shift appears when moving from lab data to deployment sites. Privacy and consent limit data sharing. Strong practice improves reliability. Use subject-wise splits to avoid leakage. Report precision, recall, F1 score, and mAP at a fixed IoU. Report hardware, FPS, and latency. Consider ATSS and quality-aware losses, such as GFL, to achieve more stable training [21, 35]. Use modern backbones and efficient attention mechanisms in computing [31, 42]. These choices align with the "Architectural Improvements" and "Relevance" columns in Table 1.

3 Research Methodology

This section describes the methodology for a systematic literature review on YOLO-based fall detection. It follows PRISMA guidelines to support clarity, transparency, and repeatability [53]. We defined a review protocol before the search. The protocol fixed the research questions, target population, and setting, outcomes, eligible study designs, time window, and analysis plan. It then guided seven steps. Inclusion criteria required vision-based fall detection using a YOLO variant, human subjects in lab or real-world contexts, standard detection metrics, English full-text access, and sufficient method detail for extraction. Exclusion criteria removed out-of-scope studies, wearable-only or ambient-only systems without vision, insufficient reporting, non-archival formats, non-English or inaccessible items, duplicates, and survey articles from quantitative synthesis. The search covered major computing and health databases. It employed keyword families, Boolean operators, and field

filters. We logged queries and counts by source, and Table 2 lists the results. Screening was followed by staged selection, with independent reviewers, conflict adjudication, and recorded reasons at each step. The workflow and criteria appear in Figures 3 and 4. Data extraction used a structured form to capture bibliographic details, YOLO variants, datasets, metrics, deployment factors, limitations, and context, as summarized in Table 3. The synthesis combines descriptive statistics with a structured narrative because datasets, splits, and reporting vary across studies. Quality was assessed using a four-criterion checklist. Two authors reviewed each paper and resolved disagreements by consensus. The assessment informed how we interpreted and reported the results. This protocol links search decisions, screening outcomes, and evidence synthesis through the referenced figures and tables.

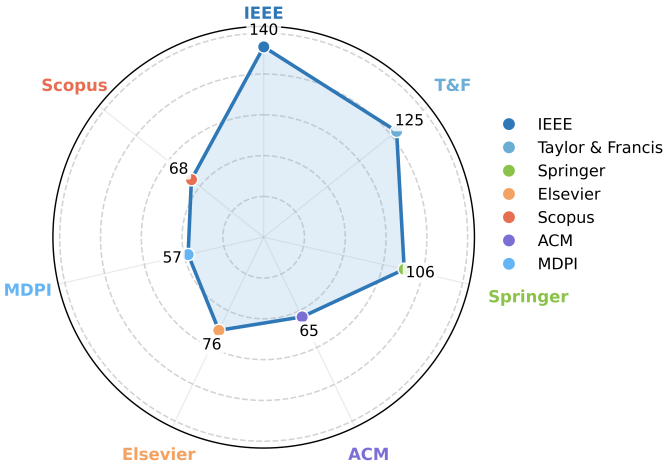

Figure 3. Search and screening process for selecting studies in the systematic literature review.

3.1 Review Protocol Development

The review protocol was drafted after we fixed the research questions and before any search began. It set objectives, scope, and boundaries for YOLO-based fall detection. It defined the population, settings, outcomes, eligible designs, time window, language, and access rules. It specified target databases, keyword families, Boolean logic, field filters, and a plan to record counts by source. It defined duplicate removal, role screening, and third-party conflict resolution. We ran a small pilot to align decisions and create eligibility notes and a codebook. Although pilot alignment and codebook development were used to improve consistency, a formal inter-reviewer agreement statistic was not recorded during screening. The protocol fixed a data extraction form with fields

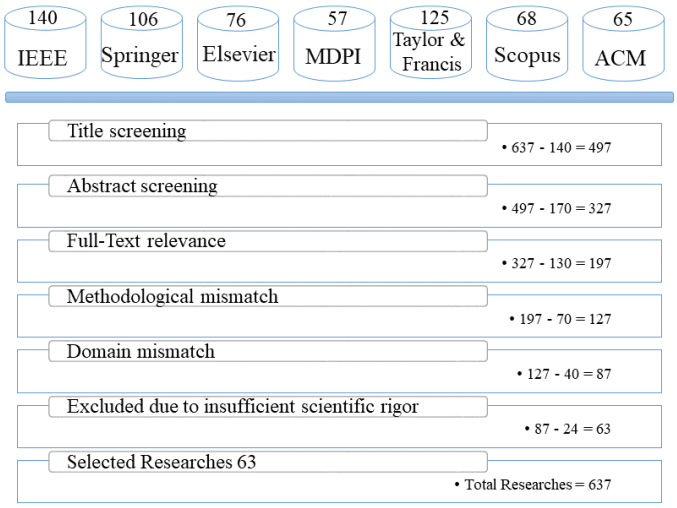

Figure 4. Exclusion criteria applied during the selection of studies in the systematic literature review.

for bibliographic details, YOLO variant, datasets, evaluation metrics, throughput, hardware, training details, and code availability. We then ran a second pilot to confirm consistent capture. A checklist-guided quality assessment was conducted across method soundness, clarity of data and design, reproducibility, and practical relevance. It used a 0-2 scoring scale and tiers for synthesis. The synthesis plan used descriptive statistics and a structured narrative because datasets and reporting vary. We did not plan meta-analytic pooling. Every step used version control, change logs, and timestamps. Any deviation required a short rationale. A reference manager supported duplicate removal. A spreadsheet supported screening and extraction; exact per-database query strings and run logs are available from the corresponding author on request. Ethics review was not required because the work uses published material. The protocol supports transparency and repeatability from search to synthesis and links each decision to a figure or table in this section.

3.2 Inclusion Criteria

We included studies that met all of the following criteria:

- **Topic scope:** Vision-based fall detection with a YOLO variant as the primary detector or as a core stage in a multi-stage pipeline. A study was considered YOLO-based when the YOLO module directly contributed to the final fall-related prediction, either as the principal detector or as an essential detection stage whose output was used in the final decision.

- **Population and setting:** Human subjects in lab or real-world environments relevant to healthcare, elder care, or home monitoring.
- **Outcomes:** Reports at least one standard detection metric, such as precision, recall, F1-score, intersection over union (IoU), or mean average precision (mAP). We recorded throughput metrics, such as frames per second (FPS), when available.
- **Study type:** Peer-reviewed journal or conference paper. We also considered preprints for YOLOv10 to YOLOv12 or for technical details not yet peer-reviewed. We mark these as preliminary and do not use them for cross-study metrics.
- **Time window:** Published between 2016 and 2025.
- **Language and access:** English language with full-text access.
- **Method detail:** Sufficient description of the YOLO architecture, dataset, and evaluation protocol to extract key variables.

3.3 Exclusion Criteria

We excluded studies that met any of the following criteria:

- **Scope mismatch:** No YOLO component or not focused on fall detection. Studies were excluded from the main YOLO-based synthesis when YOLO was used only for auxiliary functions, such as preprocessing, region proposal, or person localization before a separate non-YOLO classifier made the final prediction.
- **Non-vision pipelines:** Wearable-only or ambient sensor-only systems without a vision module. We may cite these papers for context.
- **Insufficient reporting:** No standard detection metrics or missing core method details needed for data extraction.
- **Document type:** Abstract-only records, posters, tutorials, workshop summaries, theses, or patents without full peer-reviewed content.
- **Language or access:** Non-English or full text not available.
- **Duplicates:** Duplicate records or superseded versions. We retained the most complete version.
- **Reviews:** Survey and review articles are excluded from quantitative synthesis. We may cite them in background sections.

3.4 Search Strategy

A keyword search was done in well-known databases, including IEEE Xplore, SpringerLink, Elsevier ScienceDirect, ACM Digital Library, Taylor & Francis Online, MDPI, Wiley Online Library, PubMed, Scopus, and Google Scholar. Search terms included: YOLO and Fall Detection, Vision-based Fall Detection and Deep Learning, Real-time Fall Detection and YOLO, YOLO and Elderly Monitoring, and Deep Learning and Fall Detection and YOLO. Boolean operators (AND, OR) and filters for year, field, and document type were employed to refine the results. Table 2 shows the number of papers found in each database. The overall search process is shown in Figure 3, and the inclusion and exclusion steps are shown in Figure 4. Searches were run in IEEE Xplore, SpringerLink, Elsevier ScienceDirect, ACM Digital Library, Taylor & Francis Online, MDPI, and Scopus with the year filter 2016-2025, English language, and article or conference document types, conducted in recent months. Exact per-database strings and run logs are available from the corresponding author on request.

3.5 Screening and Selection

Two reviewers independently screened titles and abstracts, with disagreements resolved by a third reviewer. Screening consistency was supported through prior pilot alignment on a random sample of 20 records before the main screening began, achieving full consensus on eligibility decisions, and through shared eligibility notes developed during protocol design. A formal inter-reviewer agreement statistic, such as Cohen's kappa, was not recorded. This is a methodological limitation; readers should consider it when interpreting the breadth and reproducibility of the screening process. The search retrieved 637 records across the target databases. Title screening removed 140 that were not about YOLO-based fall detection, leaving 497 for abstract review. Abstract screening removed 170 that lacked a YOLO model, a vision pipeline, or a fall outcome, leaving 327 for full-text assessment. Full-text checks removed 130 because the scope was off-topic, the article was a survey or tutorial, the document was not peer-reviewed, or the text was unavailable, leaving 197. Methodological scope checks removed 70 that did not perform object detection or used non-YOLO methods, leaving 127. Domain suitability checks removed 40 that targeted other health events or generic activity recognition, leaving 87. Quality appraisal removed 24 for missing

datasets, incomplete metrics, weak baselines, or unclear protocols. We included 63 studies. Counts match Figures 3 and 4.

3.6 Data Extraction and Synthesis

A simple data extraction form was made to collect key points from each selected study. Databases included *IEEE Xplore*, *SpringerLink*, *Elsevier ScienceDirect*, *ACM Digital Library*, and *Taylor & Francis Online*. Information recorded included author names, year, and where the paper was published. The YOLO model type (YOLOv3, YOLOv4, YOLOv5, YOLOv9) was noted. The datasets used were listed, including URFD, SDUFall, Le2i, DiverseFall, CAUCAFall, and custom datasets. Metrics like accuracy, precision, recall, F1-score, and IoU were recorded. Deployment factors such as real-time use, edge device support, and hardware requirements were also noted. Challenges such as lighting changes, occlusion, and class imbalance were included. A summary of extracted items is in Table 3. Quantitative pooling across studies was not feasible because included studies differed substantially in IoU thresholds (ranging from 0.25 to 0.75 across studies), evaluation units (per-frame, per-instance, or per-event), confidence thresholds, and dataset partitioning strategies. These sources of heterogeneity preclude direct metric aggregation and are the primary reason this review adopts a structured narrative synthesis rather than meta-analytic pooling.

3.7 Quality Assessment

We conducted a qualitative quality assessment of all 63 included studies using a checklist to understand how clearly and reliably each paper reported its methods and results. The aim was to highlight strengths and limitations that affect how much confidence can be placed in the findings, especially when comparing evidence across different datasets and experimental settings.

Our assessment focused on four key aspects:

- **Method soundness:** We examined whether the experimental design was appropriate for fall detection, including the use of suitable data splits, steps to avoid data leakage, and consistent evaluation practices. We also checked whether studies addressed class imbalance and reported key performance measures clearly.
- **Clarity of data and study design:** We checked whether the dataset and data collection process were described in sufficient detail, such as dataset size, class balance, camera placement, frame rate, and collection protocol. We also looked for a clear definition of a fall, annotation procedures, privacy considerations, and the splitting strategy.
- **Reproducibility:** We assessed whether the paper provided enough information to reproduce the approach, including model and training details, key hyperparameters, and evaluation procedures. When code or trained models were not shared, we noted whether the description was still detailed enough to replicate the pipeline.
- **Relevance:** We verified whether the work was directly aligned with YOLO-based fall detection and whether it reported outcomes meaningful for practical use, such as safety-relevant measures, real-time feasibility, or evidence supporting deployment in home or clinical environments.

All 63 studies were retained in the review. The quality assessment was used to guide the interpretation of results and to qualify conclusions where reporting limitations or methodological weaknesses could affect the strength of evidence.

4 Results and Analysis

This section analyzes the included studies to answer the review questions. It covers four areas. It summarizes how YOLO versions change over time. It describes the datasets used for fall detection. It lists the evaluation metrics used in prior work. It compares performance across YOLO versions. Each section concludes with a table for quick reference. We highlight consistent patterns and repeated gaps across studies. We use these findings to indicate where the evidence is strong and where further work is needed. This section also links reported results to deployment limits, including real-time use and operation on resource-limited devices.

4.1 YOLO Variants and Architectural Evolution

YOLO has evolved from YOLOv1 to YOLOv12, and this progression directly affects its suitability for visual fall detection. Figure 5 summarizes the main architectural developments across versions and the evidence scope considered in this review. In fall detection, these changes matter because the task requires robust handling of occlusion, sensitivity to small or partially visible persons near the floor, and efficient real-time inference. YOLOv1 [24] introduced unified single-pass detection through a regression-based design. Comprehensive reviews of YOLO architectural

Table 4. Comparison of datasets used in YOLO-based fall detection studies.

Dataset	Modalities	Fall Classes	Realism	Public Availability	Strengths and Limitations
UR Fall [54]	RGB-D	Yes	Moderate	Yes	Depth supports separation from clutter; limited subject and scene diversity.
Le2i [55][a]	RGB	Yes	High	Yes	Home-like scenes with variation; lacks depth and has relatively few fall instances.
NTU RGB+D [56]	RGB-D, Skeleton, Infrared	Yes	Moderate	Yes	Large and multi-stream; not dedicated to falls and needs mapping and preprocessing.
CAUCAFall [57]	RGB	Yes	High	Yes	Uncontrolled home setting with lighting changes and occlusions; protocols vary across studies.
DiverseFall [50, 58]	RGB	Yes	High	Yes	Diverse scenes and viewpoints; some releases provide YOLO-ready labels; sample balance can vary.
Synthetic Datasets [15]	Simulated video	Yes	Variable	Varies	Scalable and configurable; domain gap relative to real environments.

[a] The Le2i dataset was originally introduced by Charfi et al. (2012, 2013); [55] is cited here as a representative study that employs and describes this dataset in the fall detection context.

evolution across versions provide useful context for tracing how these foundational design choices have been adapted across domains [63]. Although it enabled real-time performance, it struggled with small targets and overlapping objects, which limited its effectiveness in cluttered indoor fall scenes. YOLOv2 and YOLO9000 [26] addressed some of these limitations by adding anchor boxes, batch normalization, and higher input resolution. Their use of Darknet-19 and joint training also improved generalization across varied rooms and camera settings. YOLOv3 [25] further improved performance through Darknet-53 and multi-scale prediction with a Feature Pyramid Network (FPN), which supports detection of smaller targets and partial views in fall scenes [25]. YOLOv4 [30] strengthened feature extraction and fusion through Cross Stage Partial (CSP) connections, Spatial Pyramid Pooling (SPP), and PANet, while augmentation methods such as CutMix and Mosaic improved robustness. These properties are relevant to fall detection studies involving multiple poses and scene settings, including NTU RGB+D [32]. YOLOv5 [33], although not an official continuation of the original series, became widely adopted because its PyTorch implementation simplified training, scaling, and deployment. This

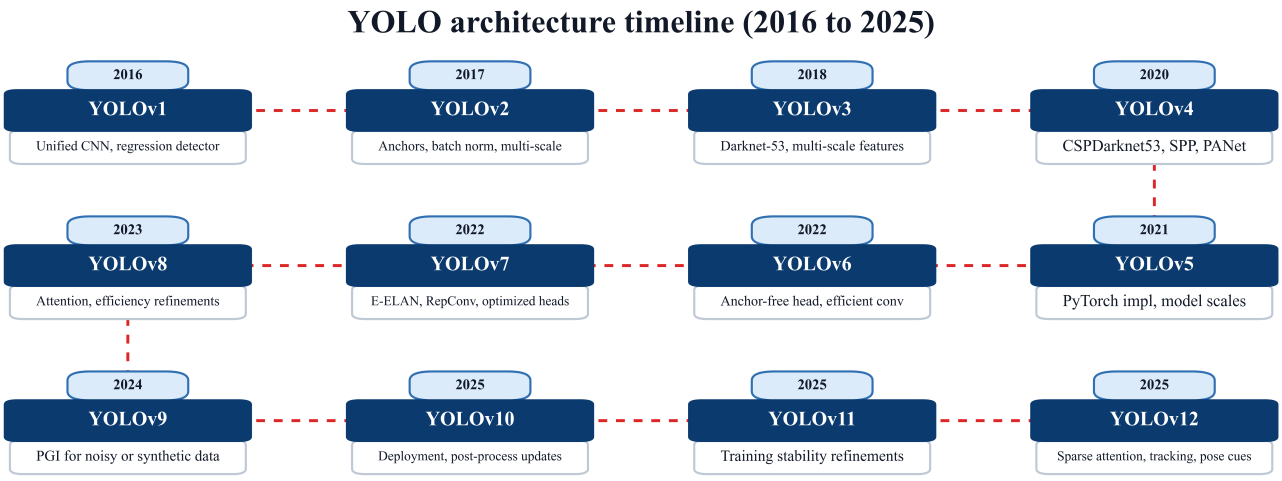

Figure 5. YOLO architecture timeline showing major versions and key design cues from YOLOv1 to YOLOv12.

practical accessibility supported its extensive use in applied fall detection [34].

YOLOv6 [36] further emphasized industrial deployment through an anchor-free head and updated label assignment. YOLOv7 [38] introduced E-ELAN and re-parameterized convolutions, and fall detection studies reported favorable trade-offs for indoor monitoring [39]. YOLOv8 [40] adopted a decoupled head and a training-friendly design, with reported gains in limited-data and multi-class fall detection settings [41]. YOLOv9 [15] introduced Programmable Gradient Information (PGI) to improve gradient flow, and related fall detection studies used synthetic or multi-modal data to examine robustness under controlled shifts [14]. Evidence for YOLOv10 to YOLOv12 in fall detection remains limited in the reviewed studies. YOLOv10 [48] reports improvements on general detection benchmarks, but fall-specific benchmark results are still sparse. YOLOv11 has been applied to Le2i, URFD, and MCFD, often with tracking or pose cues [46, 47], but its fall detection evidence is still less mature than that of earlier versions. For YOLOv12, support remains preliminary and is mainly based on sparse preprints and related real-time detector studies rather than established fall-specific evaluations [52]. Overall, YOLO shows a clear progression toward better scale sensitivity, stronger feature fusion, and more deployment-aware design, although fall detection validation remains uneven across versions.

4.2 Datasets Used in YOLO-based Fall Detection

Dataset choice strongly influences how well a YOLO-based fall detector generalizes beyond controlled settings. This subsection addresses RQ2 by summarizing the datasets used in prior studies and their main limitations. Greater variation in rooms, lighting, camera placement, and subject behavior generally improves external validity, whereas small or staged datasets can inflate reported performance and obscure failure modes. The UR Fall Detection dataset is one of the earliest vision datasets for fall detection [54]. It contains Kinect-based RGB-D sequences of falls and daily activities. Its depth modality helps separate the body from background clutter, but limited subject and scene diversity may constrain generalization. The Le2i dataset provides RGB videos recorded with static cameras in home-like environments; it was originally introduced for fall detection research and has been widely adopted in subsequent studies [55]. It includes falls and daily activities across different rooms and viewpoints, although it lacks depth and contains relatively few fall instances.

NTU RGB+D is primarily an action recognition dataset that includes fall-related actions [56]. Its RGB, depth, skeleton, and infrared streams support multimodal modeling, but fall-specific use requires additional preprocessing for label mapping and event definition. CAUCAFall focuses on fall recognition in uncontrolled home environments and includes lighting variation and occlusions [57]. This makes it useful for robustness testing, although evaluation protocols remain inconsistent across studies. DiverseFall is a newer dataset family that emphasizes diversity in camera angles, environments, and subjects [50, 58]. Some versions provide labels in YOLO-ready formats, which reduces setup effort, but class imbalance and limited sample counts can still restrict interpretation. Synthetic datasets offer scalable control over pose, lighting, and camera placement, and they help address the rarity of fall events. However, the domain gap remains a major challenge, so results from synthetic data still require validation in real scenes. This issue remains important for newer detector families, including YOLOv9 [15], where transfer from staged or synthetic data to real-world environments remains an open problem. Table 4 summarizes the datasets discussed in this subsection, including modality, realism, availability, strengths, and limitations.

4.3 Evaluation Metrics

Evaluating YOLO-based fall detection needs two views. The first view is the classification of fall versus non-fall. The second view is the localization quality for the detected person. We use standard detection terms. True positives (TP) are correct fall detections. False positives (FP) are alarms when no fall occurs. False negatives (FN) are missed falls. True negatives (TN) are correct non-fall cases. Precision in Equation (1) is the fraction of correct alarms. Recall in Equation (2) is the fraction of real falls detected. In care settings, recall reduces missed events, and precision reduces alarm load. The F1-score in Equation (3) balances both and is particularly informative under class imbalance, where accuracy alone can be misleading; its relationship to precision, recall, and other evaluation measures is formally analyzed in the literature [59]. Report the confidence threshold used to compute these scores, for example, 0.50. Also, report the evaluation unit. Some studies score per frame. Others score per person instance. Others score per event over a video segment.

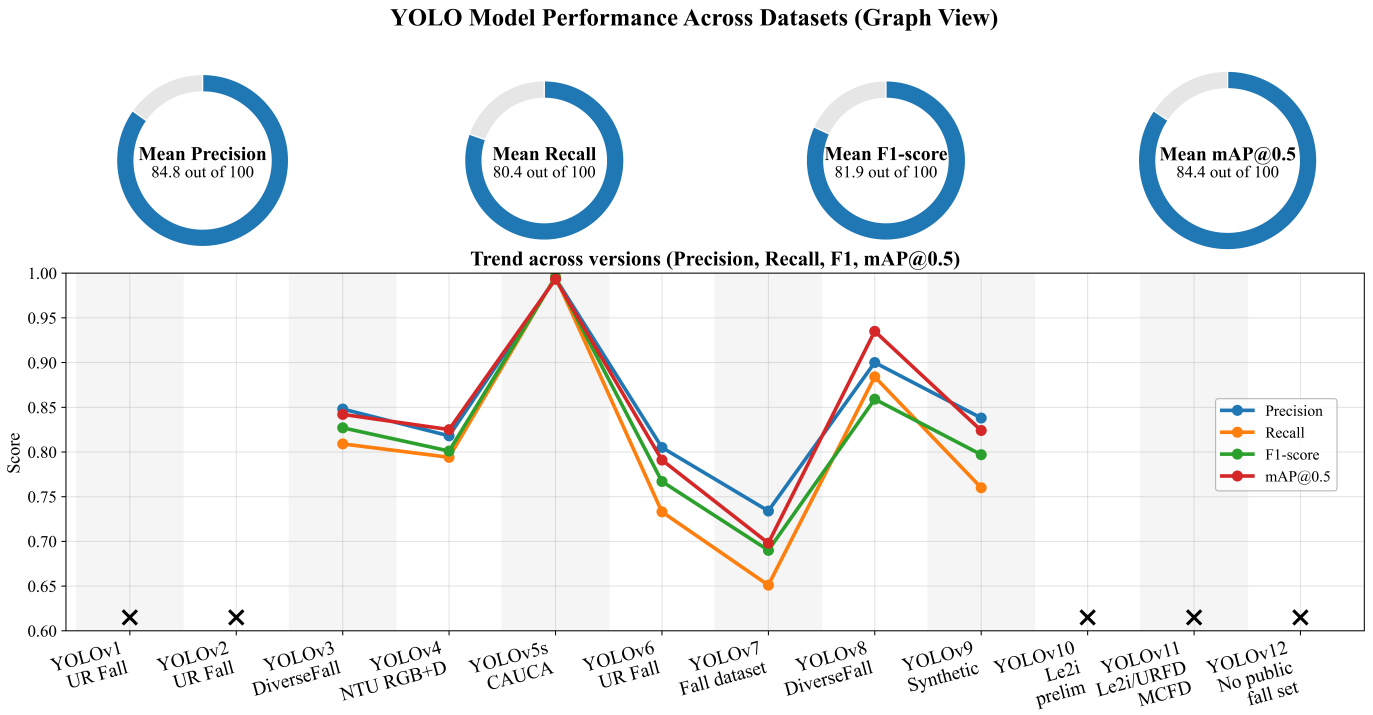

Figure 6. Graph view of YOLO performance across datasets. The four donut charts report mean Precision, mean Recall, mean F1-score, and mean mAP@0.5 over rows with available metrics. The line plot shows per-version trends for Precision, Recall, F1-score, and mAP@0.5; missing values are marked as N/A.

State the unit so results are comparable. Where useful, include the precision-recall curve and the average precision (AP) value. Prefer precision-recall over receiver operating characteristic (ROC) curves under class imbalance. If ROC is reported, include the area under the curve (AUC) and state its limits for rare events.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{F1-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

Mean average precision (mAP) in Equation (4) averages AP across classes. Many fall studies use a single-person class, so report AP and state the class set. Also, report the matching policy and the non-maximum suppression (NMS) setting used to remove duplicate boxes [60].

Alongside accuracy, report computation and latency for real-time use. Frames per second (FPS) reports throughput. Latency reports the end-to-end delay in milliseconds. For deployment-oriented interpretation, we consider a system to be real-time when it sustains approximately video-rate inference on the target hardware, typically around 24 to 30 FPS, or when it reports comparably low end-to-end latency suitable for continuous monitoring. Because reported speed depends strongly on hardware, model scale, and runtime conditions, FPS and latency should always be interpreted together with the stated device and evaluation setup. Parameter count and model size report memory needs. Floating-point operations (FLOPs) per inference report the compute cost. Unless otherwise noted, we report speed on the source GPU with batch size 1 and 640×640 input. For edge devices, also report power draw in watts and peak memory in megabytes. For video-level evaluation, report the detection delay (in seconds) from fall onset to the first alarm and the false alarms per hour; embedded platform implementations demonstrate that these temporal metrics are critical for assessing practical fall detection systems [61].

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{4}$$

Table 5. YOLO model performance across datasets.

Model	Dataset	Prec.	Rec.	F1	mAP@0.5	Reference
YOLOv1	N/A	N/A	N/A	N/A	N/A	[24]
YOLOv2	N/A	N/A	N/A	N/A	N/A	[26]
YOLOv3	DiverseFall	0.848	0.809	0.827	0.842	[25, 50]
YOLOv4	NTU RGB+D	0.818	0.794	0.801	0.825	[30, 50]
YOLOv5s	CAUCAFall	0.9943	0.9961	0.9954	0.9932	[33, 50]
YOLOv6	UR Fall	0.805	0.733	0.767	0.791	[15, 36]
YOLOv7	Public human fall dataset	0.734	0.651	0.690	0.698	[15, 38, 58]
YOLOv8s(CBAMs)	DiverseFall	0.900	0.884	0.859	0.935	[40, 50, 58]
YOLOv9	Synthetic	0.838	0.760	0.797	0.824	[15]
YOLOv10	Le2i (preliminary, small-scale)	N/A	N/A	N/A	N/A	[48]
YOLOv11	Le2i/URFD/MCFD (applied)	N/A	N/A	N/A	N/A	[46, 47]
YOLOv12	N/A (no public fall set)	N/A	N/A	N/A	N/A	[52]

Note: N/A = no harmonized fall-dataset metrics available in the cited fall-detection literature for that version. YOLOv1, YOLOv2, YOLOv10, YOLOv11, and YOLOv12 are retained for architectural continuity but excluded from quantitative comparison and graph averages. Unless stated, precision, recall, and F1 follow the reporting protocol of the cited source; mAP uses IoU 0.50. The near-perfect YOLOv5s result on CAUCAFall (Prec. 0.9943, Rec. 0.9961) reflects study-specific single-split conditions without cross-dataset validation and should not be interpreted as expected performance in real deployment settings.

4.4 Performance Comparison of YOLO Variants

Fall detection performance depends on three linked factors. The dataset defines realism, subject diversity, and camera geometry. The scene illustrates how difficult the frames are, with changes in angle, clutter, and lighting. The device defines what you can run in real time. We identify patterns within each dataset and then compare models. We treat recall as the safety driver because missed events matter most in care settings. We read precision in that light to understand alarm load. Frames per second depend on the test platform; therefore, all speed claims require the reported hardware for context. In this review, real-time feasibility is interpreted only when reported speed is tied to the target hardware and deployment context; otherwise, throughput claims are treated as indicative rather than conclusive for practical use. The three tables organize results, capacity, and feature support so you can select a version that fits your data and device. Figure 6 visualizes these trends by summarizing mean metrics and showing how precision, recall, F1-score, and mAP@0.5 evolve across versions where results are available.

Table 5 reports precision, recall, F1 score, and mean average precision at 0.5 IoU for each model and dataset pair, which shows how performance shifts with data realism and architecture strength. Results from staged or highly controlled datasets should be interpreted as evidence of baseline capability under limited variability, whereas stronger robustness claims require validation on more realistic datasets with greater scene complexity, subject diversity, and environmental variation. YOLOv1 and YOLOv2 have no harmonized fall metrics here. DiverseFall varies the actors and rooms. YOLOv3 achieves 0.848 precision, 0.809 recall, 0.827 F1, and 0.842 mAP@0.5, reflecting stronger features and better use of context. NTU RGB+D scales up scenes and motions. YOLOv4 achieves 0.818 precision, 0.794 recall, 0.801 F1, and 0.825 mAP@0.5, which are consistent with improvements in aggregation and training stability. CAUCAFall includes elderly subjects and scenes with occlusions. YOLOv5s achieves a balance of 0.9943 precision, 0.9961 recall, 0.9954 F1, and 0.9932 mAP@0.5, which is a useful point for clinical settings. In UR Fall, YOLOv6 achieves 0.805 precision, 0.733 recall, 0.767 F1, and 0.791 mAP@0.5, indicating reported fall-detection capability under the selected protocol. The public human fall dataset provides another benchmark setting. YOLOv7 achieves 0.734 precision, 0.651 recall, 0.690 F1, and 0.698 mAP@0.5, showing performance under this reported public dataset protocol. On DiverseFall, YOLOv8s(CBAMs) achieves 0.900 precision, 0.884 recall, 0.859 F1, and 0.935 mAP@0.5, which aligns with improved scale handling and stronger heads.

Synthetic sets allow clean labels and repeatable lighting. YOLOv9 achieves 0.838 precision, 0.760 recall, 0.797 F1, and 0.824 mAP@0.5, but plans that rely on synthetic gains require validation on homes and wards to mitigate domain shift. YOLOv10 to YOLOv12 transitions appear in early work using non-standard protocols and small samples. Fall metrics are unavailable; treat these rows as capability signals rather than settled results for deployment or procurement.

5 Conclusion and Future Work

This review examined 63 peer-reviewed studies on YOLO-based fall detection and followed a transparent PRISMA-guided protocol. In this review, the evidence is strongest for the progression from YOLOv3 to YOLOv9 on public and reported fall-detection datasets. The studies report clear gains over earlier vision pipelines for real-time use. Recent models improve throughput and reduce latency while maintaining strong detection quality in smart homes, hospitals, and elder care facilities. The most consistent gains appear with YOLOv5, YOLOv8, and YOLOv9. Multi-scale features, decoupled heads, and improved training recipes raise both precision and recall. Performance still depends on data realism and camera setup. Larger and more diverse datasets, such as NTU RGB+D and DiverseFall, support stronger generalization. Synthetic datasets support stress testing and controlled ablations, but require careful validation before deployment in homes and wards. Studies that report precision, recall, F1-score, mAP@0.5, and latency give a clearer readiness signal than accuracy alone, especially under strict memory and power limits. Systems that encode posture cues and handle occlusion report greater stability in cluttered rooms, which aligns with needs in clinical and assisted living settings. We treat YOLOv1, YOLOv2, and YOLOv10 to YOLOv12 as context only because current reports rely on missing, preliminary, or non-harmonized fall evaluations. Future work should turn these trends into dependable practice. First, benchmark YOLOv10 to YOLOv12 on public fall datasets using fixed protocols, shared thresholds, and matched split rules. Pair accuracy reports with device reports for GPUs and on-device neural accelerators. Next, combine frame-level detection with temporal reasoning so systems reduce missed events and suppress alarm bursts. Short motion histories, pose trajectories, and simple state machines provide practical starting points. Strengthen domain adaptation via semi-supervised learning on deployment videos to ensure models remain stable across rooms, days, and cameras. Treat privacy as a design constraint. Favor on-device processing, reduce resolution or use silhouette views when feasible, and publish retention and access policies that users can audit. For edge use, report quantization, pruning, and distillation, along with accuracy, end-to-end latency, and memory and power consumption. New ideas such as token pruning and sparse attention look promising, but they need controlled comparisons on fall datasets before adoption.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

Habib Khan served as an Associate Editor of the *ICCK Transactions on Sensing, Communication, and Control* at the time of manuscript submission. To ensure the integrity of the peer-review process, Habib Khan was not involved in the editorial handling, peer review, or decision-making process for this manuscript, which was handled independently by another editor. The remaining authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] World Health Organization. (2021). *Falls*. WHO Newsroom Fact sheets. Retrieved from https://www.who.int/news-room/fact-sheets/detail/falls

[2] Noury, N., Fleury, A., Rumeau, P., Bourke, A. K., Laighin, G. O., Rialle, V., & Lundy, J. E. (2007, August). Fall detection-principles and methods. In *2007 29th annual international conference of the IEEE engineering in medicine and biology society* (pp. 1663-1666). IEEE. [CrossRef]

[3] Vaiyapuri, T., Lydia, E. L., Sikkandar, M. Y., Díaz, V. G., Pustokhina, I. V., & Pustokhin, D. A. (2021). Internet of things and deep learning enabled elderly fall detection model for smart homecare. *IEEE Access*, *9*, 113879-113888. [CrossRef]

[4] Bourke, A. K., & Lyons, G. M. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical engineering & physics, 30*(1), 84-90. [CrossRef]

[5] Kepski, M., & Kwolek, B. (2014, August). Detecting human falls with 3-axis accelerometer and depth sensor. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 770-773). IEEE. [CrossRef]

[6] Gutiérrez, J., Rodríguez, V., & Martin, S. (2021). Comprehensive review of vision-based fall detection systems. *Sensors, 21*(3), 947. [CrossRef]

[7] Alam, E., Sufian, A., Dutta, P., & Leo, M. (2022). Vision-based human fall detection systems using deep learning: A review. *Computers in biology and medicine, 146*, 105626. [CrossRef]

[8] Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., & Moya-Albor, E. (2019). A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset. *Computers in biology and medicine, 115*, 103520. [CrossRef]

[9] Priadana, A., Nguyen, D. L., Vo, X. T., Choi, J., Ashraf, R., & Jo, K. (2025). HFD-YOLO: Improved YOLO Network Using Efficient Attention Modules for Real-Time One-Stage Human Fall Detection. *IEEE Access, 13*, 41248-41258. [CrossRef]

[10] Zi, X., Chaturvedi, K., Braytee, A., Li, J., & Prasad, M. (2023). Detecting human falls in poor lighting: object detection and tracking approach for indoor safety. *Electronics, 12*(5), 1259. [CrossRef]

[11] Zeng, G., Zeng, B., & Hu, H. (2023). Real-world efficient fall detection: Balancing performance and complexity with FDGA workflow. *Computer Vision and Image Understanding, 237*, 103832. [CrossRef]

[12] Khalili, S., Mohammadzade, H., & Ahmadi, M. M. (2022). Elderly fall detection using CCTV cameras under partial occlusion of the subjects body. *arXiv preprint arXiv:2208.07291*. [CrossRef]

[13] Schneider, D., Marinov, Z., Baur, R., Zhong, Z., Düger, R., & Stiefelhagen, R. (2025). OmniFall: A Unified Staged-to-Wild Benchmark for Human Fall Detection. *arXiv preprint arXiv:2505.19889*. [CrossRef]

[14] Denkovski, S., Khan, S. S., Malamis, B., Moon, S. Y., Ye, B., & Mihailidis, A. (2022). Multi visual modality fall detection dataset. *IEEE Access, 10*, 106422-106435. [CrossRef]

[15] Huang, X., Li, X., Yuan, L., Jiang, Z., Jin, H., Wu, W., ... & Bai, H. (2025). SDES-YOLO: A high-precision and lightweight model for fall detection in complex environments. *Scientific Reports, 15*(1), 2026. [CrossRef]

[16] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). [CrossRef]

[17] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125). [CrossRef]

[18] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8759-8768). [CrossRef]

[19] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 113-123). [CrossRef]

[20] Tan, M., Pang, R., & Le, Q. V. (2020, June). EfficientDet: Scalable and Efficient Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10778-10787). IEEE. [CrossRef]

[21] Zhang, S., Chi, C., Yao, Y., Lei, Z., & Li, S. Z. (2020, June). Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9756-9765). IEEE. [CrossRef]

[22] Khekan, A. R., Aghdasi, H. S., & Salehpour, P. (2024). The impact of YOLO Algorithms within fall detection application: A review. *IEEE Access, 13*, 6793-6809. [CrossRef]

[23] Kangas, M., Konttila, A., Lindgren, P., Winblad, I., & Jämsä, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture, 28*(2), 285-291. [CrossRef]

[24] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788). [CrossRef]

[25] J. Redmon and A. Farhadi (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. [CrossRef]

[26] Redmon, J., & Farhadi, A. (2017, July). YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 6517-6525). IEEE. [CrossRef]

[27] Raza, A., Yousaf, M. H., & Velastin, S. A. (2022, June). Human fall detection using YOLO: a real-time and AI-on-the-edge perspective. In *2022 12th International Conference on Pattern Recognition Systems (ICPRS)* (pp. 1-6). IEEE. [CrossRef]

[28] Zhang, J., Wu, C., & Wang, Y. (2020). Human fall detection based on body posture spatio-temporal evolution. *Sensors, 20*(3), 946. [CrossRef]

[29] Hussain, M. (2024). Yolov5, yolov8 and yolov10: The

go-to detectors for real-time vision. *arXiv preprint arXiv:2407.02988*. [CrossRef]

[30] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. [CrossRef]

[31] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11976-11986). [CrossRef]

[32] Gaya-Morey, F. X., Manresa-Yee, C., & Buades-Rubio, J. M. (2024). Deep learning for computer vision based activity recognition and fall detection of the elderly: a systematic review. *Applied Intelligence, 54*(19), 8982-9007. [CrossRef]

[33] Jocher, G., Stoken, A., Chaurasia, A., Borovec, J., Kwon, Y., Michael, K., ... & Thanh Minh, M. (2021). ultralytics/yolov5: v6. 0-YOLOv5n'Nano'models, Roboflow integration, TensorFlow export, OpenCV DNN support. *Zenodo*. [CrossRef]

[34] Chen, T., Ding, Z., & Li, B. (2022). Elderly fall detection based on improved YOLOv5s network. *IEEE Access, 10*, 91273-91282. [CrossRef]

[35] Li, X., Wang, W., Hu, X., Li, J., Tang, J., & Yang, J. (2021, June). Generalized Focal Loss V2: Learning Reliable Localization Quality Estimation for Dense Object Detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11627-11636). IEEE. [CrossRef]

[36] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., ... & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*. [CrossRef]

[37] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021, June). RepVGG: Making VGG-style ConvNets Great Again. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13728-13737). IEEE. [CrossRef]

[38] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023, June). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7464-7475). IEEE. [CrossRef]

[39] Zhao, D., Song, T., Gao, J., Li, D., & Niu, Y. (2024). Yolo-fall: A novel convolutional neural network model for fall detection in open spaces. *IEEE Access, 12*, 26137-26149. [CrossRef]

[40] Sohan, M., Sai Ram, T., & Rami Reddy, C. V. (2024). A review on yolov8 and its advancements. In *International conference on data intelligence and cognitive informatics* (pp. 529-545). Springer, Singapore. [CrossRef]

[41] Sanjalawe, Y., Fraihat, S., Abualhaj, M., Al-E'Mari, S. R., & Alzubi, E. (2025). Hybrid deep learning for human fall detection: A synergistic approach using YOLOv8 and time-space transformers. *IEEE Access, 13*, 41336-41366. [CrossRef]

[42] Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems, 35*, 16344-16359. [CrossRef]

[43] Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*. [CrossRef]

[44] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... & Weller, A. (2020). Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*. [CrossRef]

[45] Bolya, D., Fu, C. Y., Dai, X., Zhang, P., Feichtenhofer, C., & Hoffman, J. (2022). Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*. [CrossRef]

[46] Tîrziu, E., Vasilevschi, A. M., Alexandru, A., & Tudora, E. (2025). Real-time fall monitoring for seniors via YOLO and voice interaction. *Future Internet, 17*(8), 324. [CrossRef]

[47] Kong, V., Soeng, S., Thon, M., Cho, W. S., Nayyar, A., & Kim, T. K. (2025). PIFR: A novel approach for analyzing pose angle-based human activity to automate fall detection in videos. *Plos one, 20*(6), e0325253. [CrossRef]

[48] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). Yolov10: Real-time end-to-end object detection. *Advances in neural information processing systems, 37*, 107984-108011. [CrossRef]

[49] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024, June). DETRs Beat YOLOs on Real-time Object Detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16965-16974). IEEE. [CrossRef]

[50] Khan, H., Ullah, I., Shabaz, M., Omer, M. F., Usman, M. T., Guellil, M. S., & Koo, J. (2024). Visionary vigilance: Optimized YOLOV8 for fallen person detection with large-scale benchmark dataset. *Image and Vision Computing, 149*, 105195. [CrossRef]

[51] Syamsul, M., & Wibowo, S. A. (2024, December). Optimizers Comparative Analysis on YOLOv8 and YOLOv11 for Small Object Detection. In *2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)* (pp. 978-983). IEEE. [CrossRef]

[52] Menaka, S. R., Kamali, R., Hariesh, R., & Vengatesh, K. (2025, August). Enhancing Accuracy in Real-Time Object Detection Using YOLOv12 Model with Transformer-Based Attention Mechanisms. In *2025 International Conference on Next Generation Computing Systems (ICNGCS)* (pp. 1-8). IEEE. [CrossRef]

[53] Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *bmj, 372*. [CrossRef]

[54] Kepski, M., & Kwolek, B. (2018). Event-driven system for fall detection using body-worn accelerometer and depth sensor. *IET Computer Vision, 12*(1), 48-58. [CrossRef]

[55] Vishnu, C., Datla, R., Roy, D., Babu, S., & Mohan, C. K. (2021). Human fall detection in surveillance videos using fall motion vector modeling. *IEEE Sensors Journal, 21*(15), 17162-17170. [CrossRef]

[56] Shahroudy, A., Liu, J., Ng, T. T., & Wang, G. (2016, June). NTU RGB+ D: A Large Scale Dataset for 3D Human Activity Analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1010-1019). IEEE. [CrossRef]

[57] Guerrero, J. C. E., España, E. M., Añasco, M. M., & Lopera, J. E. P. (2022). Dataset for human fall recognition in an uncontrolled environment. *Data in brief, 45*, 108610. [CrossRef]

[58] Zaghden, N., Ibrahim, E., Safaldin, M., & Mejdoub, M. (2025). Integrating Attention Mechanisms in YOLOv8 for Improved Fall Detection Performance. *Computers, Materials & Continua, 83*(1). [CrossRef]

[59] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*. [CrossRef]

[60] Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019, June). Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 658-666). IEEE. [CrossRef]

[61] Kwolek, B., & Kepski, M. (2014). Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer methods and programs in biomedicine, 117*(3), 489-501. [CrossRef]

[62] Luo, Z., Jia, S., Niu, H., Zhao, Y., Zeng, X., & Dong, G. (2024). Elderly fall detection algorithm based on improved YOLOv5s. *Information Technology and Control, 53*(2), 601-618. [CrossRef]

[63] M.A.R. Alif and M. Hussain (2024). YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain. *arXiv preprint arXiv:2406.10139*. [CrossRef]

Babar Zeb holds a Bachelor's degree in Computer Science from the Institute of Management Sciences, Peshawar, and a Master's degree in Software Engineering from the College of Electrical and Mechanical Engineering at NUST, Islamabad. He is currently working as a researcher with a focus on computer vision, machine learning, and artificial intelligence. His research interests include developing intelligent systems for real-world applications such as fall detection, human activity recognition, and AI-powered robotics. He is passionate about using cutting-edge technologies to solve practical problems and advance smart, adaptive systems by integrating deep learning, embedded computing, and IoT. (Email: babar.zeb16@ce.ceme.edu.pk)

Muhammad Talha Usman is a Master's student at Gachon University, South Korea, and a Research Assistant in the CIP Research Group. He received a B.S. degree in Software Engineering from the University of Engineering and Technology Taxila (UET), Pakistan, in 2022. His research interests focus on Computer Vision and Pattern Recognition, Image Processing, Generative Adversarial Networks (GANs), and Graph Neural Networks (GNNs). (Email: talhausman@ieee.org)

Habib Khan is an AI Researcher and Coordinator in the CIP research network at Gachon University. He actively engages in international collaborative projects within the AI community, fostering innovation and knowledge exchange. His research interests span Machine Learning and Deep Learning, with a specific focus on Visual Intelligence in Surveillance, Saliency Detection, Object Detection, Segmentation, AI in Healthcare, Cybersecurity, Underwater Robotics, and Energy Analytics (Energy Consumption and Generation Prediction). He has contributed many scientific articles to prestigious international journals and conferences, and some of his articles are in the pipeline for Q1, top-10, A* venues. (Email: habibkhan@ieee.org)

Alexandros Gazis is a software engineer and researcher with extensive experience in distributed systems, game-based learning, AI, middleware architectures, and intelligent data processing. His work focuses on cloud and IoT middleware, context-aware systems, serious games, and performance-aware architectures. He has contributed to several international research projects and peer-reviewed publications, combining applied engineering with academic research. His interests also include AI-driven system optimization and smart environments. (Email: agazis@teemail.gr)

Stylianos Pappas is a researcher and academic contributor with expertise in computer science and information systems. His research interests include artificial intelligence applications, data analytics, intelligent software systems, high-voltage transmission, electric load forecasting (both long- and short-term), wind speed prediction, and electrical insulation materials. He has participated in interdisciplinary research activities and has co-authored scientific publications in international venues. His work emphasizes practical implementations of emerging digital technologies. (Email: steliospappas@teemail.gr)

Muhammad Faizan Omer is a researcher in medical artificial intelligence with a specialized focus on deep learning and computer vision methodologies. He is currently pursuing a Master of Science in Computer Science at the National University of Sciences and Technology (NUST), where he concentrates on medical imaging and advanced segmentation techniques. His research focuses on developing precise, efficient, and clinically applicable AI-driven models for medical image analysis. With a strong emphasis on methodological rigor and evidence-based evaluation, his work aims to advance intelligent healthcare technologies through computational innovation and structured research practices. (Email: faizanomer2000@gmail.com)

Nasir Rahim is a medical AI researcher with extensive academic and research experience. He serves as a postdoctoral researcher at Gachon University, South Korea, where he works on multimodal medical data analysis for cerebrovascular and cardiovascular diseases using advanced AI. His research integrates imaging modalities such as MRA and CTA with patient health records to develop robust, clinically relevant diagnostic tools. Prior to this role, he completed his Ph.D. at Sungkyunkwan University, South Korea, where he focused on explainable AI for the detection of progression in neurodegenerative diseases using multimodal neuroimaging data. His academic training also includes a master's degree from Sejong University, South Korea, where he specialized in applying machine learning techniques to speech data in emergency environments. (Email: nrahim3797@ieee.org)