scientific reports

OPEN

An efficient algorithm for pedestrian fall detection in various image degradation scenarios based on YOLOv8n

Jianhui Xun[1]✉, Xuefeng Wang[2], Xiufang Wang[1], Xiaoliang Fan[1], Peishuai Yang[1] & Zhifei Zhang[3]

With the growing elderly population, especially in urban areas, pedestrian fall detection for the elderly has become a critical global concern. Existing pedestrian fall detection systems often suffer from low accuracy and poor performance under challenging conditions such as rain, snow, nighttime, or camera obstructions. To address these issues, this paper proposes an enhanced pedestrian fall detection algorithm called Pedestrian Chain-of-Thought Prompted Adaptive Enhancer YOLO (PCE-YOLO), based on YOLOv8n. Several improvements were made to YOLOv8n, including the integration of a Chain-of-Thought Prompted Adaptive Enhancer (CPA-Enhancer) module to boost detection performance in complex environments. Additionally, the Cross Stage Partial Bottleneck with 2 Convolution Block (C2f) was optimized to reduce computational load and parameter count without compromising performance, while the Inner Extended Intersection over Union (Inner-EIoU) loss function was employed to improve bounding box regression accuracy and speed. To validate the model's effectiveness, a dataset of 7,782 pedestrian fall images was collected, and three degraded image datasets were generated to simulate real-world conditions. PCE-YOLO improved the mean Average Precision (mAP) by 4.52% compared to YOLOv8n on both original and degraded datasets, respectively. Moreover, it achieved a frame per second (FPS) rate of 210.5, making it suitable for real-time detection applications. The results demonstrate that PCE-YOLO significantly enhances detection accuracy and speed in various challenging environments, offering a robust solution for real-time pedestrian fall detection.

Keywords Pedestrian fall detection, YOLOv8n, CPA-enhancer, Image degradation

Accidental falls are a common occurrence in people's daily lives, especially among the elderly. As individuals age, some may lose the ability to seek help after a fall, which can delay timely medical assistance and lead to serious health complications. Therefore, detecting falls promptly and providing immediate rescue actions are crucial for improving the well-being and safety of vulnerable populations. The development of pedestrian fall detection technology offers a promising solution to this issue by enabling the automatic identification of falls, allowing faster response times.

With rapid advancements in artificial intelligence (AI), deep learning techniques are being widely applied in object detection tasks, where the goal is to identify and locate objects in images or video frames. Object detection has evolved through several algorithmic advancements. For example, the Region-Based Convolutional Neural Networks (R-CNN) series, including R-CNN[1], Fast R-CNN[2], Faster R-CNN[3], and Mask R-CNN[4], uses a two-stage process. First, candidate regions are identified in the image, and then these regions are classified using deep neural networks. While this approach is accurate, it can be computationally expensive and slower due to the two-step process.

On the other hand, the "You Only Look Once" (YOLO) series of algorithms, such as YOLOv1[5], YOLOv2[6], YOLOv3[7], YOLOv4[8], and YOLOv8[9], adopts a single-stage detection process. This means that the algorithm predicts both the category and the location of objects directly in a single network pass, making it much faster and more suitable for real-time applications. However, this speed often comes with a trade-off in accuracy compared

[1]Department of Electronic Information Engineering, Jining Polytechnic, Jining 272000, China. [2]School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China. [3]School of Cyber Science and Engineering, Qufu Normal University, Qufu 273165, China. ✉email: xunjh_sci@163.com

to the two-stage R-CNN methods. The YOLOv5 model[10], for instance, introduced several improvements to enhance both speed and accuracy, using techniques like Cross Stage Partial Networks (CSPDarknet53)[11] for efficient feature extraction. YOLOv8, released in 2023, further improved the balance between speed and precision, making it suitable for tasks like detection, classification, and segmentation.

Despite the progress in pedestrian fall detection technology, most existing algorithms are designed for ideal environments, where lighting is adequate and the visual field is clear. For instance, Liu et al.[12] and Erdem et al.[13] proposed early methods for pedestrian fall detection, but both approaches suffer in complex or cluttered backgrounds. Koldo de Miguel et al.[14] proposed a low-cost vision-based pedestrian fall detection system for elderly individuals, integrating background subtraction, Kalman filtering, and optical flow algorithms. The system runs on a Raspberry Pi, achieving high detection accuracy with a focus on practical applications in smart home environments. Miao et al.[15] resented a vision-based dataset for pedestrian fall detection, advancing research related to pedestrian fall detection using vision-based approaches. Feng et al.[16] proposed an improved pedestrian fall detection model based on YOLOv5, incorporating an enhanced SENet attention mechanism and Soft-NMS to reduce missed detections and improve detection accuracy, achieving superior performance in real-world pedestrian fall detection tasks. More recent methods, such as the posture-based pedestrian fall detection by Chen et al.[17] and skeleton-based activity recognition by Yadav et al.[18], have improved performance in well-lit, controlled settings but struggle with image degradation caused by low-light, fuzzy, or occlusion. These are common challenges in real-world applications, particularly in home environments with varying lighting conditions or where parts of the body may be obscured.

Given these limitations, it becomes critical to develop an algorithm that not only performs well in ideal conditions, but also maintains high accuracy in degraded visual environments. Our study seeks to bridge this gap by proposing the PCE-YOLO algorithm, an enhancement of YOLOv8n. By optimizing the model for detecting falls in low-light, fuzzy, and occlusion scenarios, we aim to improve the reliability of pedestrian fall detection systems in real-world settings. The primary objective of our research is to ensure timely and accurate pedestrian fall detection under diverse image conditions, which has been a significant limitation of previous models. The main contributions of this paper are as follows.

- First, this paper introduces the CPA-Enhancer[19] (Chain-of-Thought Prompted Adaptive Enhancer) module based on the YOLOv8n model. For image degradation scenarios such as low light, foggy weather, rain, snow, and occlusion, the CPA-Enhancer dynamically analyzes and adapts to image degradation using CoT prompts, improving the multi-degradation of information in images under unknown degradation scenarios. This effectively alleviates the low detection accuracy of the YOLOv8n model when facing complex scenes.
- Second, to make the model lighter, we reduce the model size and computation requirements, and facilitate better deployment on terminal devices, we were inspired by the ParametersNet proposed by Kai Han et al.[20]. We introduced this new network structure to reduce the parameter count of ParametersNet and replaced the Bottleneck module in the original C2f. Although we successfully decreased the parameter count in ParametersNet, integrating this structure into the C2f module of YOLOv8n resulted in an overall increase in its parameter count while adding almost no extra FLOPs. We named the new C2f module Pm-C2f.
- Third, this paper replaces CIoU with Inner-EIoU[21] as the bounding box regression loss function. Inner-EIoU increases the loss by incorporating the distance between centers and combining it with the minimum enclosing box size, enabling the model to achieve more accurate target localization across various scales. In the pedestrian fall detection scenario, due to the angle variation in acquiring images, the shape and size of the detected targets often vary greatly. Therefore, the model is more suitable for using Inner-EIoU.

The structure of the remainder of this paper is outlined as follows: The section titled Methodology elucidates the technical intricacies of our proposed approach. Following that, the outcomes and analysis stemming from our experiments are delineated in Experimental Results and Analysis. Lastly, we offer the concluding remarks in the section titled Conclusion.

Methodology

Baseline model

The YOLOv8n algorithm is a fast single-stage object detection method, primarily consisting of four key components: Backbone, Neck, Head, and Loss Function. YOLOv8n replaces the Cross Stage Partial Network with 3 Convolution Block (C3) structure with the C2f structure based on the YOLOv5 model, combining the advantages of the Efficient Layer Aggregation Network (ELAN) structure from YOLOv7. This replacement reduces one standard convolutional layer, resulting in a more lightweight model. Additionally, leveraging the Bottleneck module enriches the gradient information of C2f. Specifically, the C3 module maintains a similar name and structure, employing three Convolution-Batch Normalization-SiLU (CBS) modules and 3 Bottleneck modules within the module. On the other hand, the C2f module is designed based on the concepts of the C3 module and ELAN. After CBS processing, the features are initially split into two parts: one part remains unchanged, while the other part undergoes processing through several Bottleneck modules. Each Bottleneck module further divides into two channels: one channel transmits the processed features to the next Bottleneck module, while the other channel preserves them for later concatenation. Finally, after passing through n Bottleneck modules, all features are fused together. The structure of the C2f block is illustrated in Fig. 1.

The neck architecture in YOLOv8n leverages a hybrid approach, combining the Feature Pyramid Network (FPN)[22] and Path Aggregation Network (PAN)[23] structures to enhance feature fusion capabilities. By incorporating both top-down and bottom-up sampling techniques, YOLOv8n ensures that high-level semantic information and precise location details are effectively integrated across multiple scales. Unlike YOLOv5, YOLOv8n omits the convolution operation during the upsampling stage. This refinement streamlines the

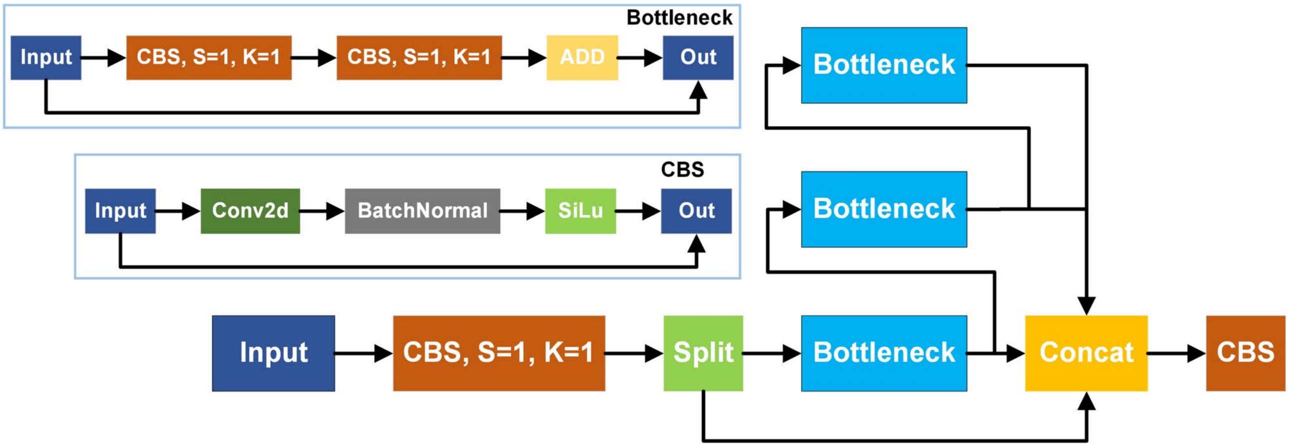


Fig. 1. Structure of C2f module.

architecture while maintaining its ability to accurately predict objects of varying dimensions. Additionally, YOLOv8n introduces a decoupled head structure inspired by the YOLOX model, further improving the fusion and utilization of feature layer information. By combining confidence and regression boxes in the final stages of the neck module, YOLOv8n achieves heightened accuracy in object detection tasks.

The detection head architecture of YOLOv8n adopts a sophisticated approach, separating the classification and detection heads as per common practice. Loss calculation is a crucial step, employing the task-aligned assigner method to discern positive and negative samples by weighing classification and regression scores. While the classification branch employs Binary Cross-Entropy (BCE) loss, the regression branch utilizes Distribution Focal Loss (DFL) and Complete Intersection over Union (CIoU) loss functions, excluding the Objectness branch. Through decoupled heads, YOLOv8n simultaneously predicts classification scores and regression coordinates, represented in matrices indicating object presence and deviation from each pixel, respectively. Additionally, a task-aligned assigner combines classification scores with Intersection over Union (IoU) values, optimizing both classification and localization while suppressing low-quality predictions. This model's decoupled head structure enhances detection accuracy and convergence, employing distinct loss functions for classification and bounding box regression tasks, thus improving model robustness and accuracy. Unlike YOLOv5, which employs a coupled head, YOLOv8n's decoupled head approach eliminates the objectness branch and transitions to an anchor-free method, focusing on target center and boundary distance estimation for precise object localization.

Improved YOLOv8n model

The improved YOLOv8n model, PCE-YOLO, incorporates two major enhancements that significantly improve its performance in degraded image scenarios. First, the CPA-Enhancer Module (Chain-of-Thought Prompted Adaptive Enhancer) is designed to handle low-light, fuzzy, and occlusion images by dynamically adjusting the model's detection capabilities through Chain-of-Thought (CoT) prompts. It uses a hierarchical encoder-decoder structure, coupled with Receptive Field Attention Convolution (RFAConv), to progressively refine feature representations and adapt the model's detection strategy based on specific degradation types, enabling more accurate detection in complex environments. Second, the Pm-C2f Module optimizes the C2f structure of YOLOv8n to make the model lightweight and efficient. By introducing a dynamic convolution-based bottleneck, Pm-C2f reduces the number of parameters and the computational load (GFLOPs) without sacrificing detection performance, making PCE-YOLO more suitable for real-time deployment on resource-constrained devices. These enhancements, combined with the Inner-EIoU loss function for more precise bounding box regression, result in a highly robust and efficient fall detection model capable of performing well under various image degradation conditions. The structure of PCE-YOLO is shown in Fig. 2.

The architecture of the PCE-YOLO system is composed of several key modules that work in concert to detect pedestrian falls efficiently under various degraded image conditions. Initially, the input image, which may be degraded by factors such as low light, blur, or occlusion, is passed through the CPA-Enhancer module. This module enhances image quality by applying a Chain-of-Thought (CoT) prompting mechanism and receptive field attention convolution (RFAConv), allowing the model to adapt to unknown image degradation scenarios and improve feature representation.

Once the image is enhanced, the optimized features are passed to the Pm-C2f module, an improvement over the YOLOv8n network. The Pm-C2f module reduces the number of parameters and computational complexity of the model, making it lightweight and suitable for real-time applications. This module efficiently extracts high-level features, ensuring that the model can process the image with minimal computational load while maintaining high detection accuracy.

Finally, enhanced features are fed into the YOLOv8n detection module, which performs object classification and bounding box regression to detect the presence of falls. The Inner-EIoU loss function is used to refine the predictions of the bounding box, ensuring precise localization of the detected targets. The combination of these

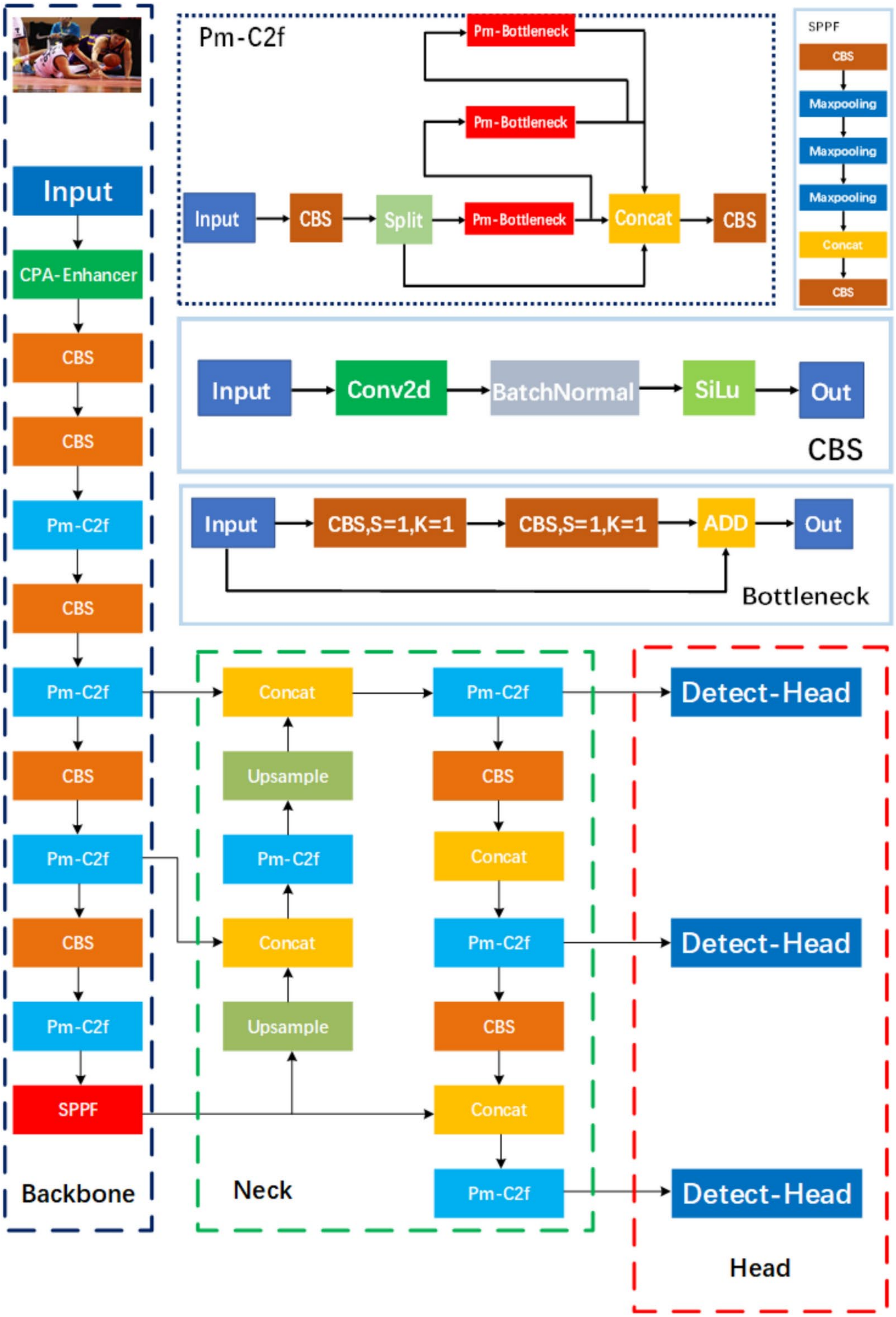


Fig. 2. Structure of PCE-YOLO.

modules results in an efficient and accurate fall detection system capable of handling real-world challenges such as image degradation.

CPA-enhancer module

The goal detection algorithm experiences varying degrees of degradation in features when dealing with image degradation scenarios such as low-light, fuzzy, and occlusion. This leads to a decrease in the detection performance of the algorithm. The CPA-Enhancer module, proposed by Yuwei Zhang et al.[19] in March 2024, is a novel approach. It serves as a plug-and-play enhancement model for target detection in degraded images, offering significant gains without prior knowledge of the degradation type. The structure of the CPA-Enhancer module introduced in this paper is illustrated in Fig. 3.

For a given image with unknown degradation, the CPA-Enhancer first extracts the low-level features F_0 using a receptive-field attention convolution (RFAConv). Subsequently, the embedded feature F_0 undergoes a 4-level hierarchical encoder-decoder, with each level passing through an RFAConv. Starting from feature F_0, the encoder gradually reduces the spatial size while generating low-resolution latent feature F_l to increase channel capacity through downsampling operations. Then, the decoder progressively restores the high-resolution feature F_h from the low-resolution latent feature F_l. Finally, F_h is processed through RFAConv to generate enhanced feature F_e, which is combined with the original image to obtain the enhanced image. This enhanced image is then fed into the YOLOv8n detector for further detection.

In the CPA-Enhancer module, a chain-of-thought prompt generation Module (CGM) is utilized to generate CoT prompts. CGM consists of stacking several transpose convolution layers and the Hardswish activation function to generate a series of multi-scale prompts. Specifically, CoT prompts are defined as Eq. (1).

$$CotP_i = Hardswish(TransConv(CotP_i)), i \in 1,2 \tag{1}$$

where $CoTP_i$ denotes the prompt generated in the *i*th layer. *TransConv* represents the transpose convolution, while Hard-Swish indicates the Hard-Swish activation function. The sizes of the decoding layers corresponding to the CoT prompts generated by CGM have varying sizes. This not only helps the model better understand degradation types in a coherent and incremental manner but also facilitates the learning of hierarchical representations by the model.

CPA-Enhancer promotes the interaction between the input feature F_i and the prompt $CoTP_i$ through the Content-Driven Prompt Block (CPB). The CPB enables the model to adjust its enhancement strategies according to the type of degradation, thus allowing the detector to perform well in various complex scenarios. The structure of the CPB is illustrated in Fig. 4. For the feature F_i output by the *i*-th decoder, Z_c^i and Z_s^i are generated through the channel-wise attention and spatial-wise attention modules, respectively, as derived from the following Eqs. (2) and (3).

$$Z_c^i = Conv_{1\times1}(RELU(Conv_{1\times1}(GAP_c(F_i)))) \tag{2}$$

$$Z_s^i = Conv_{7\times7}([GAP_s(F_i), GMP_s(F_i)]) \tag{3}$$

where GAP_c is the global average pooling operation over spatial dimensions, GAP_s is the global average pooling operation for cross-channel attention, and GMP_s is the global maximum pooling operation for cross-channel attention. $[\cdot]$ denotes the operation of concatenating $GAP_s(F_i)$ and $GMP_s(F_i)$ channel by channel. Following this, the computations of Z_c^i, Z_s^i, and F_i result in F_w^i. F_w^i is processed through a channel shuffle operation and a 7×7 stride depthwise separable convolution, then activated by a Sigmoid function to obtain F_s^i. F_s^i is combined with $CoTP_i$, which has undergone bilinear interpolation to produce F_p^i. Here, we assume that F_p^i has *C* channel dimensions and is divided into *n* blocks along *C*, with each block processed through a Transformer block to obtain F_t. F_t is derived using the following Eqs. (4) and (5).

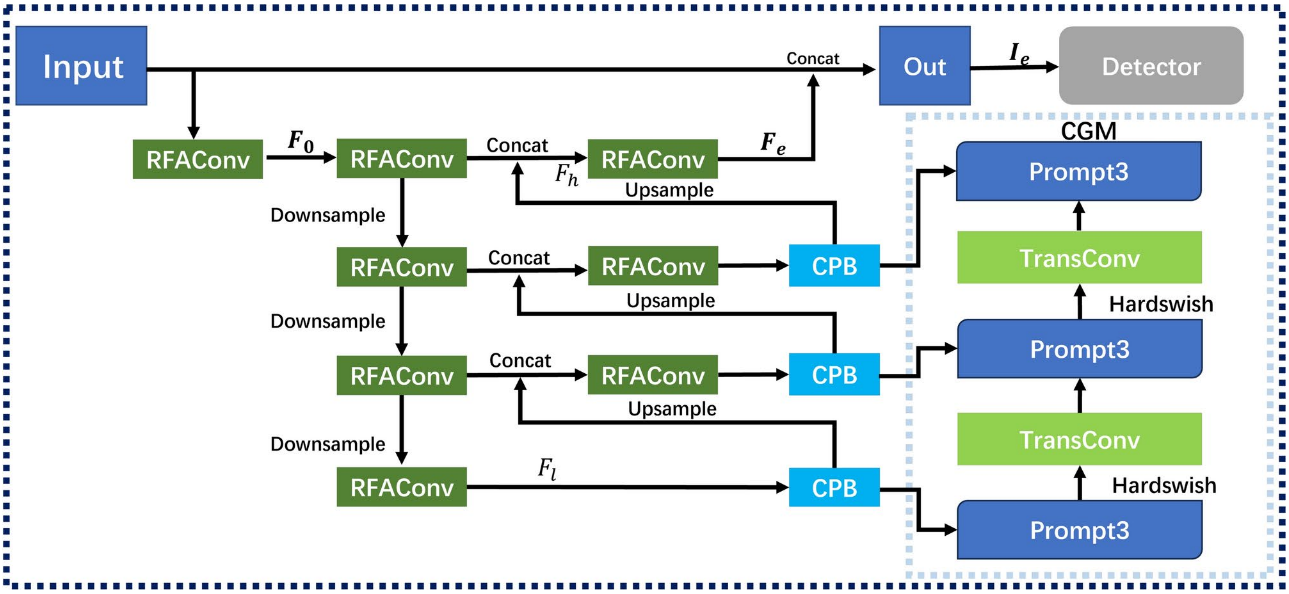


Fig. 3. Structure of CPA-Enhancer Module.

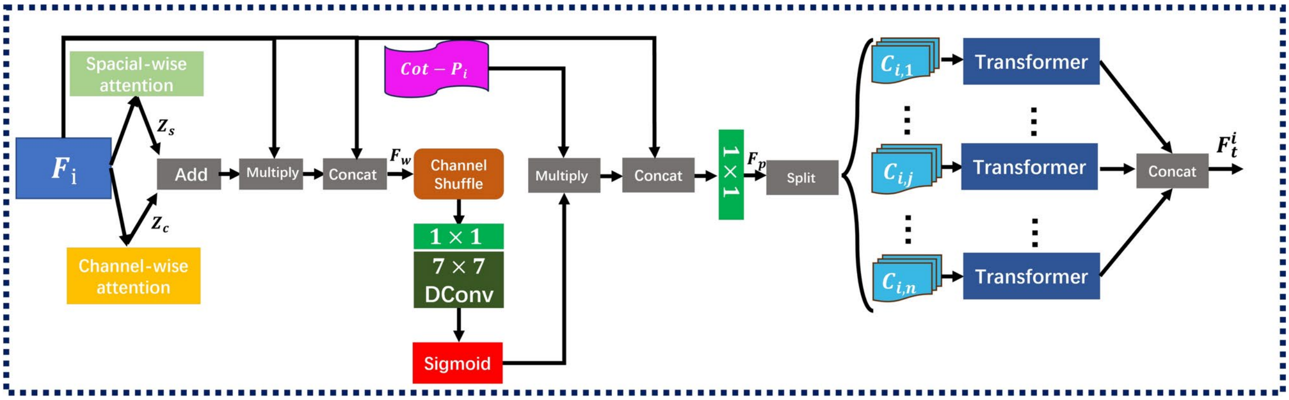


Fig. 4. Structure of Content-driven Prompt Block (CPB).

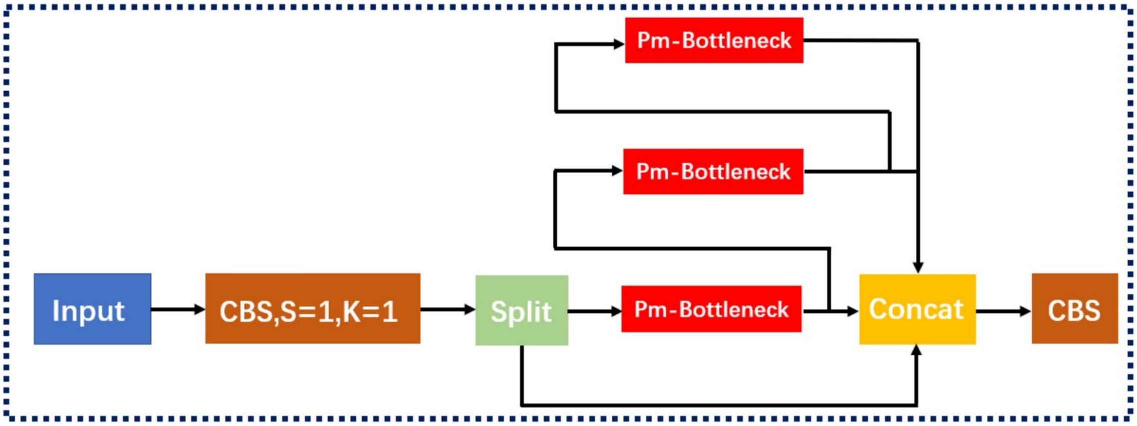


Fig. 5. Structure of Pm-C2f.

$$F_p^{i,j} = F_P^i[:, :, (j-1)\frac{C_i}{n} : j\frac{C_i}{n}] \tag{4}$$

$$F_t^{i,j} = PC(V \cdot \sigma(K \cdot Q/\alpha)) + F_p^{i,j} \tag{5}$$

where n is the total number of transformer blocks and $j \in \{1, 2, \ldots, n\}$. *PC* is point-wise convolution, and α represents a learnable scaling parameter. Then, $F_t^{i,j}$ is transformed through the Gated-Dconv Feed Forward Network (GDFN). Finally, we concatenate $F_t^{i,j}$ along the channel dimension, and the final output of CPB, F_t^i, is obtained through the following Eq. (6).

$$F_t^i = F_t^{i,1}, F_t^{i,2}, \ldots, F_t^{i,n} \tag{6}$$

Pm-C2f

After adding the CPA-Enhancer module, the YOLOv8n network demonstrates excellent performance in processing images that exhibit degradation phenomena. However, the CPA-Enhancer module significantly increases the FLOPs of the YOLOv8n model. Fall detection typically needs to be deployed on some mobile devices, and thus the addition of the CPA-Enhancer module undoubtedly increases the difficulty of deployment. To mitigate this issue, we have made lightweight improvements to the C2f module of YOLOv8n. Kai Han et al.[20] proposed a new network structure called ParametersNet based on the addition of Dynamic Convolution to the Ghost Module, which increases the model parameters without adding extra FLOPs. Inspired by this, we introduced this new network structure to reduce the number of parameters and replaced the original Bottleneck module in the C2f structure. We named the new Bottleneck module Pm-Bottleneck. The improved C2f results are shown in Fig. 5. The improved C2f in YOLOv8n contains only 2.2 million parameters and a computational cost of 5.8 GFLOPs. Furthermore, in subsequent experiments, we verified that the performance of the improved C2f module is not weakened compared to the original C2f.

The improved part of C2f, Pm-Bottleneck, is shown in Fig. 6. Pm-Bottleneck replaces the two CBS modules in the original Bottleneck with Dynamic Batch ReLU (DBR) modules. When data is passed to the DBR module, it first goes through a Dynamic Convolution[24] layer, followed by a BatchNorm layer and a ReLU activation

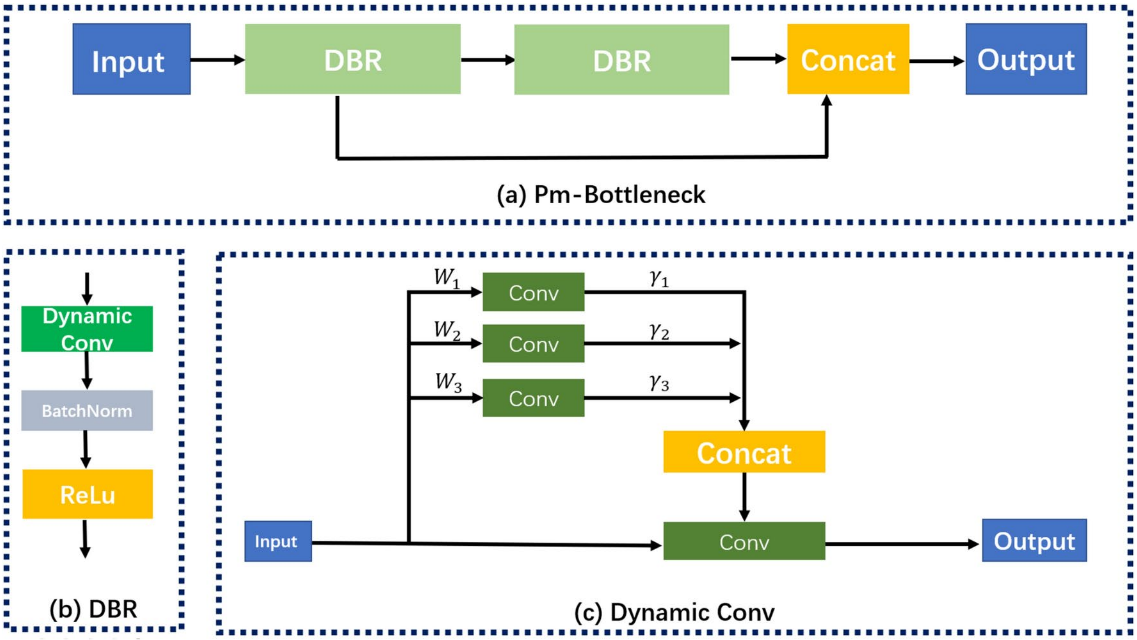


Fig. 6. The components of the Pm-C2f module.

function layer. Dynamic Convolution enhances model performance by aggregating multiple convolutional kernels through attention mechanisms. For different images, different kernels can be dynamically assembled, thus improving the model's performance.

For a given input feature map X and the weight matrix W for each convolution, Dynamic Convolution (Dynamic Conv) first needs to compute the dynamic coefficients γ_i corresponding to each matrix. The calculation process for γ_i is shown in Eq. (7). For different input feature maps X, the input information is first fused into a vector using global average pooling. Then, a two-layer MLP module with a softmax activation function is used to dynamically generate the coefficients.

$$\gamma = softmax(MLP(Pool(X))) \tag{7}$$

After calculating the dynamic coefficient γ, the dynamic convolution calculation process with N dynamic experts is shown in Eqs. (8) and (9).

$$Y = X \times W' \ , \tag{8}$$

$$W' = \sum_{i=1}^{N} \gamma_i \times W_i \tag{9}$$

Inner-EIoU

In the process of pedestrian fall detection, issues such as target occlusion, numerous interferences, and small detection targets may arise. Therefore, improving the loss function can effectively enhance the model's performance. The YOLOv8n model uses DFL combined with CIoU[25] Loss to calculate the regression localization loss of the bounding box. CIoU provides a more comprehensive and accurate evaluation of the bounding box position by considering IoU, center point distance, and aspect ratio. This allows the model to better adjust the position and shape of the predicted boxes during training, thus improving the detection accuracy. However, CIoU does not account for the balance between hard and easy samples, and its aspect ratio describes relative values. In some cases, the weight of the aspect ratio might lead to excessive penalties, making it difficult for the model to converge to the optimal position. Particularly when detecting targets with significant shape changes, the aspect ratio term may introduce additional instability. Moreover, for nearly non-overlapping bounding boxes, the IoU part of CIoU may cause gradient vanishing, affecting the model's learning effectiveness. Therefore, we use Inner-EIoU to replace the original CIoU in YOLOv8n.

Compared to other bounding box regression loss functions, EIoU not only considers IoU, center point distance, and aspect ratio but also adds an aspect ratio consistency constraint. Therefore, EIoU consists of three components: IoU loss (L_{IoU}), distance loss (L_{dis}), and aspect ratio loss (L_{asp}). By introducing these additional constraints, EIoU can more comprehensively assess the match between the predicted box and the ground truth box, effectively improving the accuracy and robustness of object detection. However, when dealing with targets that have significant aspect ratio differences, especially when detecting small targets or targets with varying shapes, EIoU may still have certain shortcomings and may introduce additional errors. The following Eq. (10) shows the detailed calculation of EIOU.

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{10}$$

where *IoU* (Intersection over Union) measures the overlap between the predicted box and the ground truth box, and $1 - IoU$ represents the *IoU* loss. $\rho(b, b^{gt})$ is the Euclidean distance between the centers of the predicted box *b* and the ground truth box b^{gt}. *h* and *w* represent the height and width of the predicted box, h^{gt} and w^{gt} denote the height and width of the ground truth box, and c_h (height) and c_w (width) represent the dimensions of the smallest enclosing box formed by the predicted and ground truth boxes.

Inner-IoU (Inner Intersection over Union) is another improved method aimed at optimizing the model's loss function by introducing the IoU of the internal region. Unlike traditional IoU, Inner-IoU focuses on the intersection area inside the predicted box and the ground truth box, thereby evaluating the overlap between targets more accurately. This method is particularly effective in handling overlapping targets and occlusion issues, reducing the impact of background noise, and improving the model's detection performance in complex scenarios. However, Inner-IoU mainly focuses on matching the internal regions and may not be as comprehensive as EIoU in optimizing the overall bounding box's position and shape. The specific calculation process of Inner-IoU is shown by the following equations and the description of Inner-IoU is shown in Fig. 7.

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} * ratio}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} * ratio}{2} \tag{11}$$

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} * ratio}{2}, b_b^{gt} = y_c^{gt} + \frac{h^{gt} * ratio}{2} \tag{12}$$

$$b_l = x_c - \frac{w * ratio}{2}, b_r = x_c^{gt} + \frac{w * ratio}{2} \tag{13}$$

$$b_t = y_c - \frac{h * ratio}{2}, b_b = y_c^{gt} + \frac{h * ratio}{2} \tag{14}$$

$$inter = (min(b_r^{gt}, b_r) - max(b_l^{gt}, b_l)) * (min(b_b^{gt}, b_b) - max(b_t^{gt}, b_t)) \tag{15}$$

$$union = (w^{gt} * h^{gt}) * (ratio)^2 + (w * h) * (ratio)^2 - inter \tag{16}$$

$$IoU^{inner} = \frac{inter}{union} \tag{17}$$

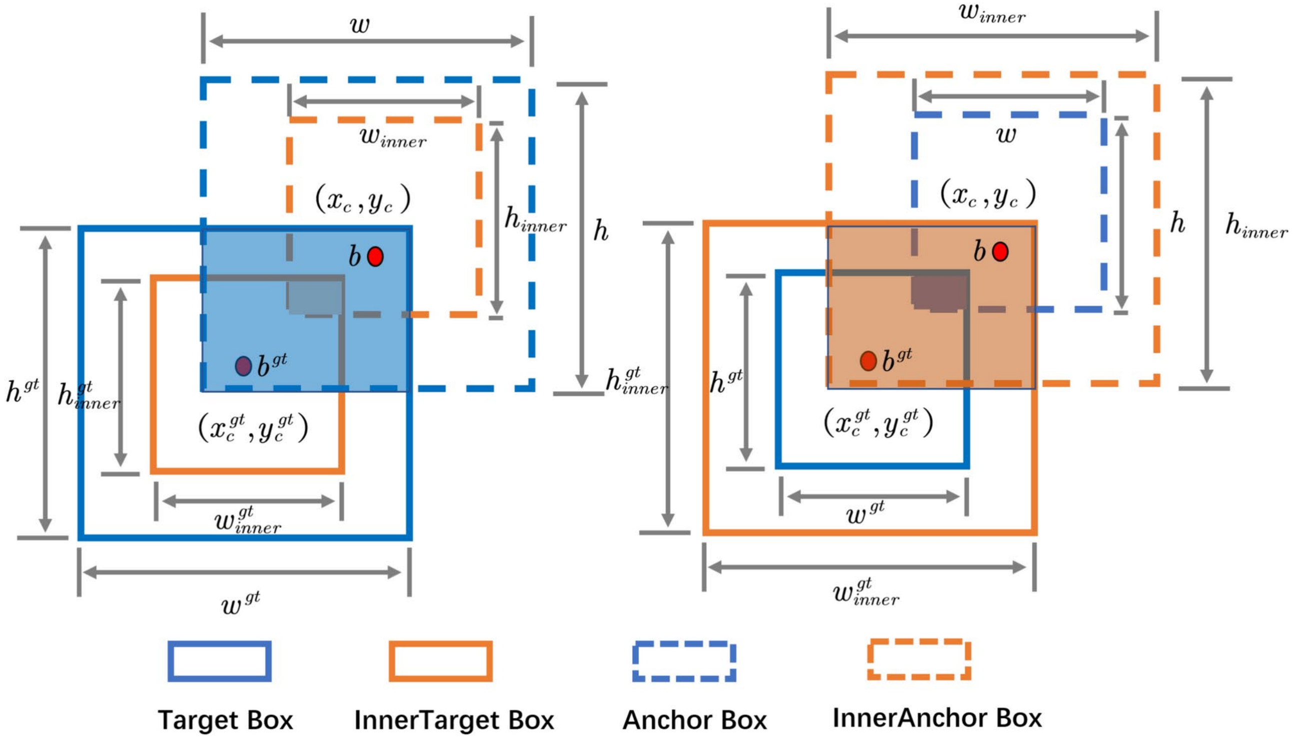


Fig. 7. Inner-IoU diagram.

As shown in Fig. 7, b^{gt} and b represent the Target Box and the Anchor Box, respectively. (x_c^{gt}, y_c^{gt}) is the center point of both the target box and the inner target box. Similarly, (x_c, y_c) is also the center point of the anchor box as well as the inner anchor box. w^{gt} and h^{gt} denote the width and height of the target box, respectively. The width and height of the anchor box are denoted by *w* and *h*, respectively. Finally, the variable ratio is the scaling factor, with a range between 0.5 and 1.5.

To combine the advantages of the above methods, we propose Inner-EIoU (Inner-Enhanced IoU), which integrates the benefits of both EIoU[26] and Inner-IoU[27]. Inner-EIoU retains the aspect ratio consistency constraints and bounding box optimization of EIoU, while incorporating the internal region evaluation of Inner-IoU. This combination allows Inner-EIoU to more accurately adjust the position and shape of the predicted boxes in challenging scenarios, such as those involving occlusion, multiple distractors, and small target detection. Additionally, Inner-EIoU dynamically adjusts the loss function weights to balance the difficulty of samples, thus enhancing the model's robustness and convergence speed. In practical applications, Inner-EIoU significantly improves the YOLOv8n model's detection accuracy and stability, providing strong support for pedestrian fall detection systems.

Experimental results and analysis

Dataset and experiments environment

In this study, we collected 7,782 images from the internet depicting pedestrian falls in various scenarios and annotated pedestrians in a fallen state. To further simulate potential complex scenes and validate model performance, we synthesized images based on the original ones that include three types of realistic scene degradation: Low-Light, Fuzzy, and Occlusion, resulting in three additional datasets. The synthesized images of the image degradation scenarios are shown in Fig. 8.

Figure 8 shows two examples from the image data. The first column displays the original images. The second column shows low-light scenes simulated by altering the brightness of the original images. The third column represents fuzzy scenes simulated by adding blur effects and white scatter. The fourth column simulates partial occlusion by adding black occlusion blocks. To verify the model's performance under different image degradation conditions, we will train and test the algorithmic model on these four different datasets.

We divided the collected 7,782 images and the synthesized images under three degradation environments, resulting in four datasets, and divided them into training, validation, and test sets at an 8:1:1 ratio. Specifically, the training set contains 80 percent of the images, while the validation and test sets each contain 10 percent of the images. We will train, validate, and test both the original and improved algorithms on these four datasets to ensure comprehensive evaluation of the model's performance across various scenarios.

We conducted the training, validation, and testing of the model on the Ubuntu 20.04.6 LTS operating system using four NVIDIA GeForce RTX 4090 GPUs in parallel. We utilized the open-source PyTorch deep learning framework along with GPU acceleration provided by NVIDIA CUDA.

The PCE-YOLO model was trained from scratch, with the first 10 epochs serving as a warm-up phase to stabilize the training process. The complete training process consisted of 120 epochs, with a momentum of 0.937

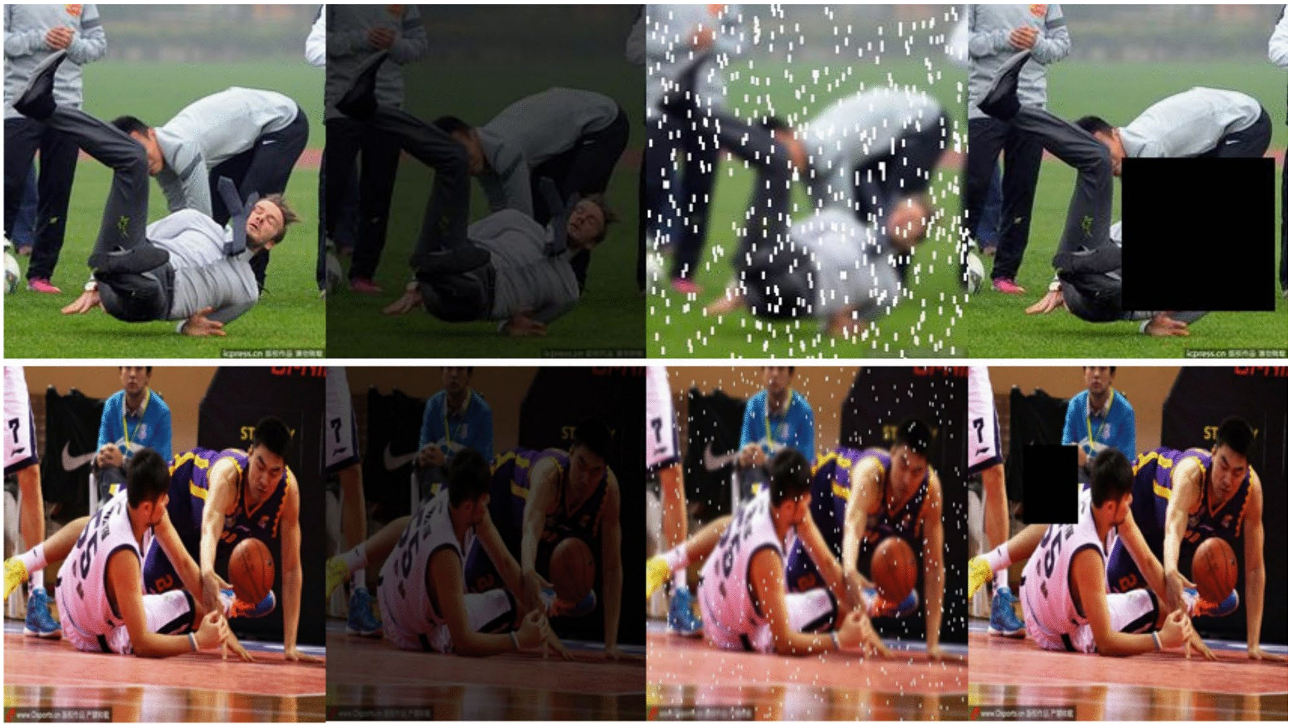


Fig. 8. Original and generated image samples in datasets.

to help balance between stability and convergence. An initial learning rate of 0.01 was used, gradually decreasing to a final learning rate of 0.0001 as training progressed. To prevent overfitting and improve generalization, a weight decay of 0.0005 was applied.

A batch size of 256 was used to efficiently utilize the computational resources, and the Adam optimizer was employed to handle parameter updates. During training, a Mosaic data enhancement strategy was applied, which was disabled in the final 10 epochs to ensure smoother convergence. The input image size was set at 640 × 640 pixels to maintain consistency across the datasets. Some key parameters of the model are shown in Table 1.

After training, the PCE-YOLO model was tested across four different datasets: the original fall detection dataset and three degraded versions simulating low-light, fuzzy, and occluded scenarios. The purpose of this testing phase was to evaluate the model's robustness and performance in various real-world conditions.

Evaluation metrics

In this paper, we selected commonly used evaluation metrics in object detection for analysis, including Precision, Recall, mAP (mean Average Precision), GFLOPs, and parameters. Among these metrics, TP (True Positive) refers to the cases where the classifier correctly predicts positive samples that are indeed positive in the actual data. TN (True Negative) refers to the correct identification of negative samples, where the predictions are negative, and they are truly negative in reality. FP (False Positive) and FN (False Negative) represent the number of false alarms and missed detections, respectively.

Precision is calculated as the ratio of TP to the sum of TP and FP, indicating the accuracy of positive predictions made by the model. The following equation (18) shows the detailed calculation of Precision.

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

Recall is the ratio of TP to the sum of TP and FN, reflecting the model's ability to identify all actual positive samples. The following equation (19) shows the detailed calculation of Recall.

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

Accuracy is the ratio of the sum of TP and TN to the total number of predictions, indicating the overall prediction accuracy of the model. The following equation (20) shows the detailed calculation of Accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

The mean Average Precision (mAP) is a comprehensive measure that averages the precision across different recall levels, providing an overall performance metric for the model. The following equation (21) shows the detailed calculation of mAP.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{21}$$

where *N* is the number of object classes, and AP_i is the Average Precision for class *i*.

GFLOPs (Giga Floating Point Operations) is a measure of the computational complexity of the model, indicating how many billion floating-point operations are required to process an image. This metric helps in understanding the efficiency and scalability of the model when deployed in real-world scenarios. Additionally, the number of parameters in the model gives an insight into the model's size and complexity, which is crucial for evaluating the feasibility of deploying the model on devices with limited computational resources.

By using these metrics, we can thoroughly evaluate the performance of the object detection models to ensure a balance between accuracy, efficiency, and resource utilization. This detailed analysis enables us to make informed decisions in selecting and optimizing models for real-world applications.

Parameters	Setup
Epochs	120
Momentum	0.937
Initial learning rate	0.01
Final learning rate	0.0001
Weight decay	0.0005
Batch size	256
Optimizer	Adam
Data enhancement strategy	Mosaic
Input image size	640 × 640

Table 1. Some key parameters set during model training

Results

Table 2 presents a comparison between the proposed PCE-YOLO algorithm and various versions of the YOLOv8, YOLOv9, and YOLOv10 algorithms across the original Fall image dataset and three degraded image datasets (Low-Light_fall, Occlusion_fall, and Fuzzy_fall). It is evident that PCE-YOLO shows significant performance improvements when handling these degraded scenarios. On the original Fall image dataset, YOLOv8n achieves an mAP (0.5) of 89.63%, and YOLOv8s reaches 91.4%. PCE-YOLO improves upon these, achieving an mAP of 92.94%, which is 3.31% higher than YOLOv8n and 1.54% higher than YOLOv8s.

In the low-light degradation scenario, PCE-YOLO achieves an mAP of 91.47%, slightly less than 1% lower than on the original Fall dataset. In comparison, YOLOv8n and YOLOv8s achieve mAP of 87.07% and 89.9%, respectively, performing worse than PCE-YOLO. The occlusion and fuzzy degradation scenarios show even more significant improvements for PCE-YOLO. On the Occlusion_fall and Fuzzy_fall datasets, YOLOv8n achieves mAP of 83.43% and 81.96%, and YOLOv8s achieves 85.11% and 84.89%. In comparison, PCE-YOLO maintains higher performance, achieving mAP of 88.14% and 87.67% for Occlusion_fall and Fuzzy_fall, representing improvements of 4.71% and 5.71% over YOLOv8n and 3.03% and 2.78% over YOLOv8s.

When comparing to YOLOv9s and YOLOv10s, PCE-YOLO also demonstrates clear advantages. On the original Fall dataset, YOLOv9s and YOLOv10s achieve mAP of 91.84% and 92.56%, while PCE-YOLO reaches 92.94%, outperforming YOLOv9s by 1.1% and YOLOv10s by 0.38%. In the low-light scenario, YOLOv9s and YOLOv10s achieve mAP of 89.42% and 91.3%, while PCE-YOLO achieves 91.47%, outperforming YOLOv9s by 2.05% and YOLOv10s by 0.17%. In the occlusion scenario, YOLOv9s and YOLOv10s achieve mAP of 85.64% and 86.17%, while PCE-YOLO reaches 88.14%, improving by 2.5% and 1.97%, respectively. In the fuzzy scenario, YOLOv9s and YOLOv10s achieve mAP of 85.71% and 86.07%, while PCE-YOLO achieves 87.67%, improving by 1.96% and 1.6%.

In conclusion, PCE-YOLO consistently outperforms YOLOv8, YOLOv9, and YOLOv10 across various degraded image scenarios, demonstrating superior robustness and detection performance.

To deploy the model on terminal platforms, we also need to compare the model's parameters, computational complexity, model size, and FPS. PCE-YOLO has 2.8M parameters, while YOLOv8n, YOLOv8s, YOLOv9s, and YOLOv10s have 3.15M, 11.16M, 7.2M, and 7.2M parameters, respectively. PCE-YOLO has the lowest number of parameters, reducing the model's complexity. In terms of computational complexity, PCE-YOLO has 18.6 GFLOPs, which is 10.5 GFLOPs higher than YOLOv8n's 8.1 GFLOPs. However, PCE-YOLO's complexity is much lower than YOLOv8s's 28.8 GFLOPs, YOLOv9s's 26.7 GFLOPs, and YOLOv10s's 21.6 GFLOPs, offering better performance with less computational demand.

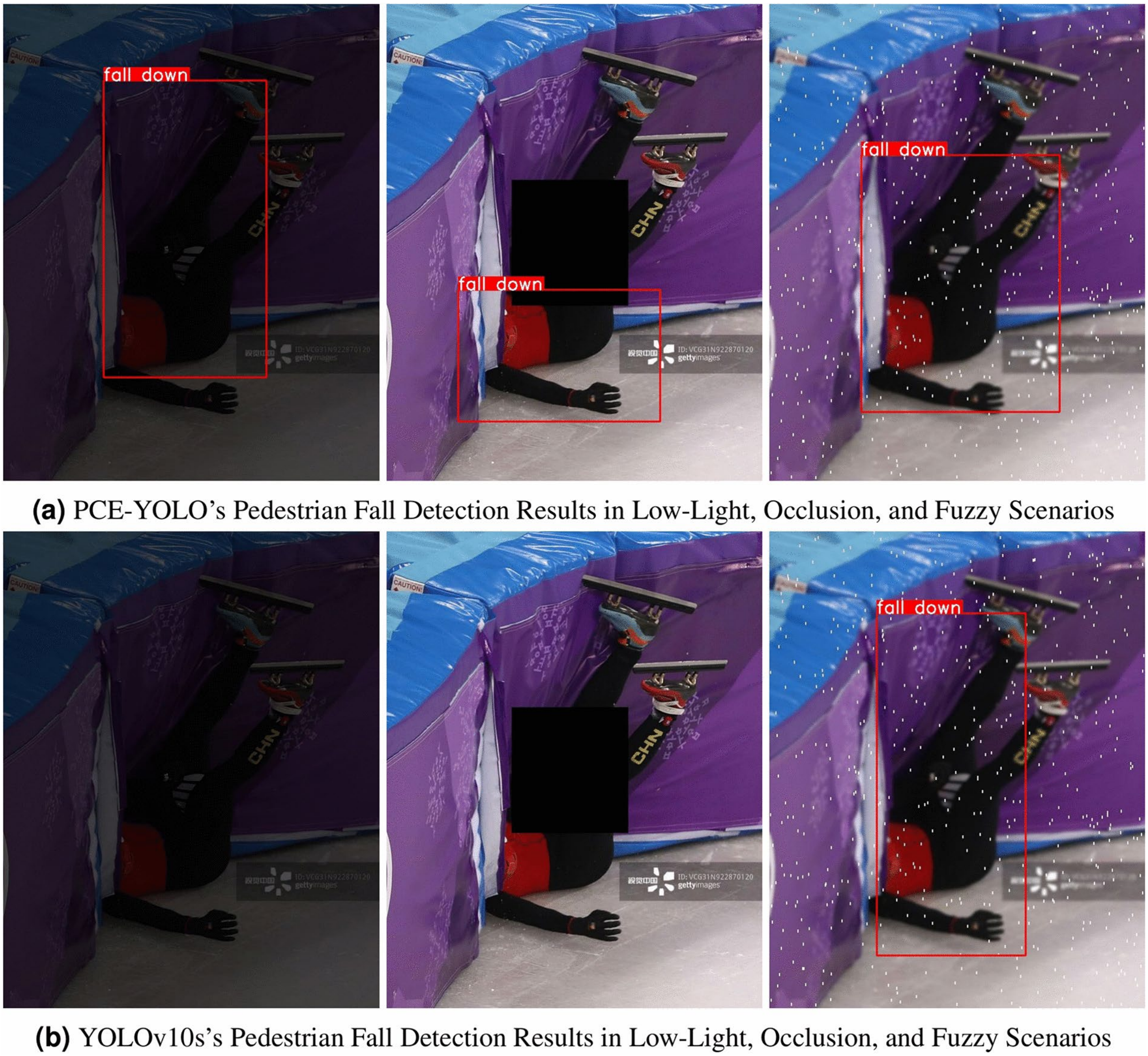
The model size of PCE-YOLO is 5.6MB, which is smaller than YOLOv8n's 6.1MB, YOLOv8s's 21.5MB, YOLOv9s's 15.9MB, and YOLOv10s's 21.5MB, making PCE-YOLO the most lightweight model and ideal for deployment on limited-resource platforms. Finally, when comparing FPS, PCE-YOLO achieves 210.5 FPS for a 640x640 image size, which is lower than YOLOv8n's 273.5 FPS, YOLOv8s's 253.7 FPS, YOLOv9s's 258.4 FPS, and YOLOv10s's 266.3 FPS. Despite this, PCE-YOLO strikes a balance between efficiency and performance, making it suitable for deployment in real-time scenarios. Table 3 provides a detailed comparison.

Datastes		Fall	Low-Light_fall	Occlusion_fall	Fuzzy_fall
Algorithms	Metrics				
PCE-YOLO	Precision/%	92.46	90.3	88.72	87.36
	Recall/%	85.97	83.23	78.13	74.53
	mAP0.5/%	**92.94**	**91.47**	**88.14**	**87.67**
	mAP0.5:0.95/%	65.18	61.01	58.63	60.1
YOLOv8n	Precision/%	89.32	88.8	83.18	81.01
	Recall/%	80.36	77.5	77.99	72.18
	mAP0.5/%	89.63	87.07	83.43	81.96
	mAP0.5:0.95/%	61.79	56.81	53.18	52.55
YOLOv8s	Precision/%	92.15	90.1	86.34	84.02
	Recall/%	83.26	81.22	79.03	78.4
	mAP0.5/%	91.4	89.9	85.11	84.89
	mAP0.5:0.95/%	62.48	58.42	56.25	55.18
YOLOv9s	Precision/%	92.71	90.47	86.79	85.2
	Recall/%	83.88	82.05	79.25	77.53
	mAP0.5/%	91.84	90.88	85.64	85.71
	mAP0.5:0.95/%	63.32	59.47	56.93	57.86
YOLOv10s	Precision/%	93.5	90.62	87.03	86.37
	Recall/%	84.1	82.5	79.41	77.79
	mAP0.5/%	92.56	91.3	86.17	86.6
	mAP0.5:0.95/%	64.4	60.62	57.06	58.42

Table 2. Comparison of PCE-YOLO and other algorithms of different scales on various datasets

Algorithm	Params/M	FLOPs/G	Size/MB	FPS
PCE-YOLO	**2.8**	18.6	**5.6**	210.5
YOLOv8n	3.15	8.1	6.1	**273.5**
YOLOv8s	11.16	28.8	21.5	253.7
YOLOv9s	7.2	26.7	15.9	258.4
YOLOv10s	7.2	21.6	21.5	266.3

Table 3. Performance comparison of PCE-YOLO and other YOLO algorithms



(a) PCE-YOLO's Pedestrian Fall Detection Results in Low-Light, Occlusion, and Fuzzy Scenarios

(b) YOLOv10s's Pedestrian Fall Detection Results in Low-Light, Occlusion, and Fuzzy Scenarios

Fig. 9. Comparison of Detection Results Between PCE-YOLO and YOLOv10s.

Figure 9 shows a comparison of pedestrian fall detection between the PCE-YOLO and YOLOv10s algorithms under different image degradation scenarios. Figure (a) illustrates the detection results of PCE-YOLO in three image degradation scenarios: Low-Light, Occlusion, and Fuzzy. It can be observed that PCE-YOLO accurately detects pedestrian falls in all degradation scenarios. Figure (b) presents the detection results of YOLOv10s, which fails to detect pedestrian falls in both the Low-Light and Occlusion degradation scenarios. Overall, PCE-YOLO demonstrates superior performance in pedestrian fall detection under degraded image conditions.

Ablation experiment results

This paper designed ablation experiments to verify the effectiveness of the proposed improvement strategies. We divided PCE-YOLO into five different models to further study the impact of each added network structure

Model	CPA-Enhancer	Pm-C2f	Inner-EIoU	mAP/%	Params/M	FLOPs/G	FPS
M1	×	×	×	85.52	3.15	8.2	273.5
M2	✓	×	×	88.65	3.51	21.0	196.4
M3	×	✓	×	86.01	2.19	5.8	288
M4	✓	✓	×	88.95	2.8	18.6	210.5
M5	✓	✓	✓	**90.04**	2.8	18.6	210.5

Table 4. Ablation experiment results of PCE-YOLO

branch on the entire model. The specific composition of the five models is as follows: the first group is the original YOLOv8n algorithm; the second group adds the CPA-Enhancer module based on the YOLOv8n model; the third group improves the C2f module on the basis of the YOLOv8n model, replacing the original C2f module with Pm-C2f; the fourth group adds the CPA-Enhancer module to the YOLOv8n model and uses Pm-C2f; the fifth group is the PCE-YOLO model proposed in this paper, which uses Inner-EIoU on the basis of the fourth group. The specific results of the ablation experiments are shown in Table 4, where M1-M5 refer to the five models mentioned above.

Table 4 shows the results of the ablation experiments. Four metrics were used to evaluate the performance differences between different model groups, with mAP representing the average results of the models across four datasets. It can be seen that the generalization ability of Model M2 improved after adding the CPA-Enhancer module, with the mAP increasing by 3.13% compared to Model M1. However, the parameter count and GFLOPs both increased significantly, and the FPS decreased to 196.4 compared to M1.Model M3 improved the C2f module based on M1. It can be observed that M3 not only enhanced the mAP compared to M1 but also significantly reduced the parameter count and GFLOPs. Additionally, the FPS increased to 288 compared to M1, showing a substantial improvement in detection speed. Model M4 achieved an mAP of 88.95, showing improvements over M1, M2, and M3, and also became more lightweight. Model M5, which incorporates all the improvement strategies, raised the mAP to 90.04%, reduced the parameter count to 2.8 compared to M1, and achieved better improvements in GFLOPs and FPS compared to M2. In conclusion, each proposed improvement strategy demonstrated significant effectiveness.

Conclusion

This paper presents a method called PCE-YOLO for recognizing pedestrian falls in various image degradation scenarios based on YOLOv8n. By introducing the CPA-Enhancer module, the original YOLOv8n's low detection performance in multiple image degradation scenarios was effectively improved. Additionally, the original YOLOv8n's low detection performance in multiple image degradation scenarios was effectively improved by introducing the CPA-Enhancer module, enhancing the C2f module, and using a new Bottleneck module, the issue of the model's increased size due to the addition of the CPA-Enhancer module was mitigated. Furthermore, we replaced the CIoU used in the original YOLOv8n with Inner-EIoU, which further improved the model's mAP. The improved model can be effectively used in various image degradation scenarios for pedestrian fall detection, addressing the low detection performance issues of algorithms in such scenarios.

The improved model was trained and tested on normal pedestrian fall image datasets and three types of degraded image datasets, and compared with the original YOLOv8n, the larger-scale YOLOv8s, as well as YOLOv9s and YOLOv10s. Experimental results show that our proposed PCE-YOLO achieved significant mAP improvements in image degradation scenarios compared to YOLOv8n, YOLOv8s, YOLOv9s, and YOLOv10s. Furthermore, with a computational requirement of 18.6 GFLOPs, PCE-YOLO is much lower than YOLOv8s's 28.8 GFLOPs, YOLOv9s's 26.7 GFLOPs, and YOLOv10s's 21.6 GFLOPs, making it well-suited for deployment on mobile and resource-constrained devices while maintaining a balance between performance and efficiency.

Data availability

The datasets used and/or analyzed during the current study are available under the dataset name 'Pedestrian Fall Detection in Various Image Degradation Scenarios Datasets': https://doi.org/10.5281/zenodo.11471987. Our original image data was sourced from https://aistudio.baidu.com/datasetdetail/127208 and is available for use under the CC BY-NC-SA 4.0 license.

Code availability

The code that support the findings of this study are available from the corresponding author upon reasonable request.

Received: 2 June 2024; Accepted: 10 March 2025
Published online: 16 March 2025

References

1. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587 (2014).
2. Girshick, R. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448 (2015).
3. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28** (2015).

4. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969 (2017).
5. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016).
6. Redmon, J. & Farhadi, A. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7263–7271 (2017).
7. Redmon, J. & Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint* arXiv:1804.02767 (2018).
8. Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint* arXiv:2004.10934 (2020).
9. Talaat, F. M. & ZainEldin, H. An improved fire detection approach based on yolo-v8 for smart cities. *Neural Comput. Appl.* **35**, 20939–20954 (2023).
10. Jocher, G. *et al.* ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo* (2022).
11. Mahasin, M. & Dewi, I. A. Comparison of cspdarknet53, cspresnext-50, and efficientnet-b0 backbones on yolo v4 as object detector. *Int. J. Eng. Sci. Inf. Technol.* **2**, 64–72 (2022).
12. Liu, H. & Hou, X. Moving detection research of background frame difference based on gaussian model. In *2012 International Conference on Computer Science and Service System*, 258–261 (IEEE, 2012).
13. Akagündüz, E., Aslan, M., Şengür, A., Wang, H. & Ince, M. C. Silhouette orientation volumes for efficient fall detection in depth videos. *IEEE J. Biomed. Health Inform.* **21**, 756–763 (2016).
14. De Miguel, K., Brunete, A., Hernando, M. & Gambao, E. Home camera-based fall detection system for the elderly. *Sensors* **17**, 2864 (2017).
15. Miao, S. et al. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. *Front. Neurorobot.* **13**, 38 (2019).
16. Feng, Y., Wei, Y., Li, K., Feng, Y. & Gan, Z. Improved pedestrian fall detection model based on yolov5. In *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 410–413 (IEEE, 2022).
17. Chen, Z., Wang, Y. & Yang, W. Video based fall detection using human poses. In *CCF Conference on Big Data*, 283–296 (Springer, 2022).
18. Yadav, S. K., Tiwari, K., Pandey, H. M. & Akbar, S. A. Skeleton-based human activity recognition using convlstm and guided feature learning. *Soft Comput.* **26**, 877–890 (2022).
19. Zhang, Y., Wu, Y., Liu, Y. & Peng, X. Cpa-enhancer: Chain-of-thought prompted adaptive enhancer for object detection under unknown degradations. *arXiv preprint* arXiv:2403.11220 (2024).
20. Han, K., Wang, Y., Guo, J. & Wu, E. Parameternet: Parameters are all you need for large-scale visual pretraining of mobile networks. *arXiv preprint* arXiv:2306.14525 (2023).
21. Ye, R., Shao, G., He, Y., Gao, Q. & Li, T. Yolov8-rmda: Lightweight yolov8 network for early detection of small target diseases in tea. *Sensors* **24**, 2896 (2024).
22. Lin, T.-Y. *et al.* Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125 (2017).
23. Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8759–8768 (2018).
24. Chen, Y. *et al.* Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11030–11039 (2020).
25. Xiao, G., Hou, S. & Zhou, H. Pcb defect detection algorithm based on cdi-yolo. *Scientific Reports* **14**, 7351 (2024).
26. Yang, Z., Wang, X. & Li, J. Eiou: an improved vehicle detection algorithm based on vehiclenet neural network. In *Journal of Physics: Conference Series*, vol. 1924, 012001 (IOP Publishing, 2021).
27. Zhang, H., Xu, C. & Zhang, S. Inner-iou: more effective intersection over union loss with auxiliary bounding box. *arXiv preprint* arXiv:2311.02877 (2023).

Author contributions

J. X. and X. W. conceived the methodology. J. X. carried out the experiments, and wrote the original manuscript. X. W. and X. F. analyzed the data and results. P. Y. and Z. Z. validated the methodology and results. J. X. and X. W. were in charge of funding, and project management. X. F. and Z. Z. were in charge of supervision. All the authors have read the manuscript and agreed to its submission for publication.

Funding

This research was funded by the High-Level Scientific Research Achievement Support Fund of Jining Polytechnic (No. JZYZ2023149) and the Double Innovation Talent Program of Jiangsu (No. JSSCRC2022266).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.X.