Efficient fall detection using lightweight network to enhance smart internet of things

Pinrolinvic D. K. Manembu, Jane Ivonne Litouw, Feisy Diane Kambey, Abdul Haris Junus Ontowirjo, Vecky C. Poekoel, Muhamad Dwisnanto Putro
Department of Electrical Engineering, Faculty of Engineering, Sam Ratulangi University, Manado, Indonesia

Article Info

Article history:

Received Aug 15, 2024
Revised May 26, 2025
Accepted Jun 30, 2025

Keywords:

Edge device
Efficient computation
Fall detection
Lightweight module
Smart internet of things

ABSTRACT

Fall detection automatically recognizes human falls, mainly to monitor and prevent severe injury and potential fatalities. It can be developed by applying deep learning methods to recognize human subjects during fall incidents and implemented in the internet of things (IoT) to monitor patient and elderly individuals' activity. The development of object detection presents you only look once v8 (YOLOv8) as an influential network, but its efficiency needs to be improved. A modified YOLOv8 architecture is proposed to introduce a novel lightweight network version called YOLOv8-Hypernano (YOLOv8h) that recognizes fall events. The backbone incorporates a combined spatial and channel attention module, which enhances focus on human subjects by concentrating on movement patterns to detect falls more accurately. This work also offers a consecutive selective enhancement (CSE) module to improve efficiency and effectiveness in feature extraction while reducing computational costs. The neck structure is modified by adding a lightweight bottleneck network. The proposed network reconstructs feature maps in depth, paying more attention to accurate human movement patterns and enhancing efficiency and effectiveness in feature extraction. Experimental results of YOLOv8h with the light bottleneck and consecutive selective enhancement modules show giga floating-point operations per seconds (GFLOPS) of 5.6 and 1,194,440 parameters. The model performance is calculated in mean average precision, achieving 0.603 and 0.732 on the Le2i and Fallen datasets, respectively. These results demonstrate that the optimized network improves accuracy performance while maintaining lightweight computing requirements that can run smoothly on IoT devices, achieving comparable speed and efficiency suitable for operation on low-cost computing devices.

Corresponding Author:

Pinrolinvic D. K. Manembu
Department of Electrical Engineering, Faculty of Engineering, Sam Ratulangi University
Bahu Campus, Manado 95115, Indonesia
Email: pmanembu@unsrat.ac.id

1. INTRODUCTION

Vision-based fall detection is a rewarding task that analyzes and predicts fall events, which can cause serious injuries such as disability, paralysis, or death. It is especially crucial for elderly patients, as falls can be fatal [1]. The risk of falls increases due to physical and mental decline influenced by aging. It makes fall detection systems essential in healthcare to improve its quality. Therefore, fall detection systems are a promising solution for reducing the risk of falls and their health consequences [2]. Meanwhile, it is impossible to prevent falls, but physical exercise and technological solutions can completely help reduce their frequency. Internet of things (IoT) utilizes this detection system to reduce risk and improve the performance of monitoring the activities of patients and the elderly [3], [4]. Besides, fall detection can enhance IoT ability by enabling continuous remote monitoring of the environment [5]. This system commonly requires edge devices with low computation cost, thus demanding a lightweight and effective algorithm. The smart IoT can implement more useful preventive measures by utilizing these detection systems [6]–[8]. It can decrease the impact of fall injuries and promote healthy and active lifestyles.

The deep learning approach has played a crucial role in extracting complex information and accurately predicting objects and behaviors [9], [10]. These networks are effective due to their non-linear operations and deep layers, which simultaneously process feature maps. The needs and challenges in computer vision emphasize convolutional neural networks (CNN) as modern methods that optimally filter out important features [11]–[14]. CNNs have proven very effective in computer vision tasks such as object recognition and image classification. The general convolution layer is followed by an activation layer to offer a non-linearity function. CNNs for fall detection have paved the way for powerful algorithms such as you only look once (YOLO). This network offers an efficient solution by detecting objects in a single process, enabling real-time object detection without sacrificing accuracy [15], [16]. YOLOv8, the latest version of the YOLO family, shows significant advancements over its predecessors. It achieves higher accuracy in object detection, making it highly reliable for applications requiring robust real-time detection [17]. However, despite its accuracy, it requires extensive energy resources. YOLOv8-nano has been presented as a more efficient algorithm but still runs slowly on devices with limited computing capacity [18]. Therefore, improving efficiency in its development is imperative to create a lighter architecture without compromising detection performance in real-world applications. Human fall detection has been extensively studied, with models designed to quickly reduce rescue times and significantly identify human body movements. However, developing a suitable architecture presents several challenges, particularly in capturing global and local information while maintaining detection accuracy. The GL-YOLO-Lite model, developed by [19], integrates a transformer block and an attention module into the YOLOv5 architecture to address these issues. The overlapping challenge in complex environments is addressed by the efficient diverse branch block-YOLO (ED-YOLO) model, which uses YOLOv5s as its backbone [20]. This research produces a real-time feature extraction that encourages the network to work optimally. Another study [8] proposed a vision-based fall detection system that employs object tracking and image enhancement techniques. Practical applications drive new research focused on presenting lightweight algorithms. A study [21] introduced a lightweight CNN architecture using YOLOv5, which replaces the entire backbone with ShuffleNetV2. A study [22] presented a method that integrates convolution and information suppression layers to reduce computational overhead while maintaining optimal detection performance. The proposed study presents an efficient solution for fall detection by introducing a resource-efficient approach, enabling implementation across multiple platforms, including IoT-based hardware, particularly edge devices.

Enhancement modules have been widely used to improve the precision of object localization [23]–[25]. The module adopted an attention mechanism designed to optimize accuracy and efficiency in this model, enhancing its ability to detect falls. This module helps the extraction feature focus on the person's body, highlighting specific attributes of movement patterns and body positions that indicate falling activity. This work focuses on integrating the enhancement module into the network's backbone to improve the precision of fall detection. The network gains additional capability in extracting essential information by effectively separating elements from the background. This advantage is achieved without adding significant computation or increasing the number of parameters.

The summary of the potential impact and contributions of this study is as follows:

a. An efficient fall detection system is developed as an IoT-based monitoring system that operates on low-cost computing devices.
b. This study proposes a consecutive selective enhancement (CSE) module that modifies the structural enhancement of YOLOv8-nano to improve fall detection performance. This modification refines the target features of the human body specific to fall events.
c. Extensive evaluation is conducted to measure the performance of the proposed detector compared to other lightweight network detectors. Additionally, the study analyzes the model's efficiency by examining the proposed model's number of parameters, computational complexity, and inference time.

2. METHOD

2.1. Backbone

In computer vision, the term "backbone" is analogous to the human backbone that supports the body. Similarly, in YOLO, the backbone is the primary foundation for CNN architecture for extracting information from input images. In YOLOv8, the C3 module from YOLOv5 has been updated to convolutional two faster (C2F). This update improves feature extraction by retaining information more quickly and efficiently. After the C2F stage, the processed output goes to the SPPF stage. This stage adds variations to the information before it moves to the neck of the YOLO architecture. Before entering the neck, a new layer called C2F-CSE is added after SPPF. It ensures that the varied information enhances the model's ability to highlight important vertical and horizontal information separately from the two spatial dimensions. This approach makes the model more focused on capturing essential features in the image. The number of channel layers is modified to reduce computation in the proposed network backbone. Limiting channel assignment encourages the network to extract features faster during the training and inference processes. A new YOLOv8 size variant called YOLOv8h (YOLOv8-Hypernano) limits the maximum number of channels in the network layer to 128, as presented in Figure 1. It significantly reduces the number of parameters and computational complexity compared to the nano version.

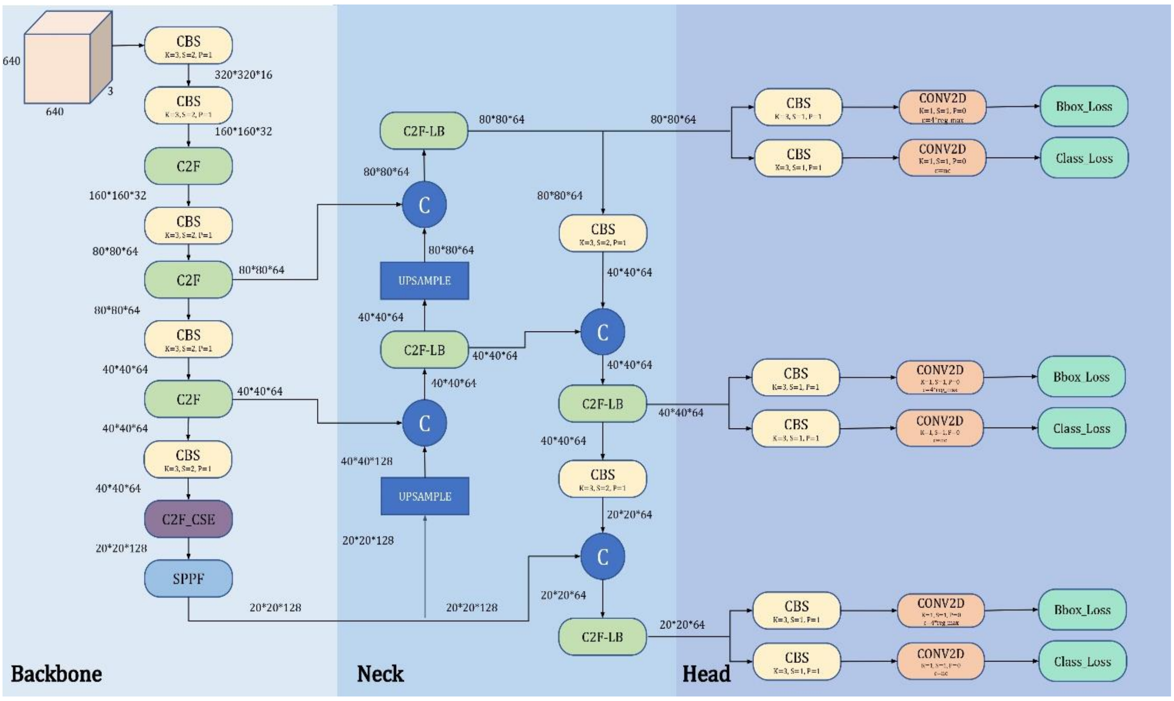


Figure 1. The proposed architecture is improved from the YOLOv8 nano version. It consists of a backbone as the main extractor feature, a neck to relate information in different frequencies, and the head is responsible for predicting the location and dimension of an object

2.1.1. C2F

The convolutional two faster (C2F) module in YOLOv8 is inspired by the previous version of the convolutional three (C3) block. This module is designed to improve model performance and efficiency in YOLOv8. The C2F module comprises two convolution operations at the network's beginning and end. The input information is split into two parts after the first convolution. The first part passes the input information by performing a residual operation. In contrast, the second part applies a bottleneck that utilizes convolutions with different kernel sizes to achieve optimal efficiency and effectiveness. Furthermore, the model combines both features to enrich the different information. At the end of the C2F module, the model's performance is enhanced more efficiently by employing a 1×1 convolution operation to consolidate the information.

2.1.2. C2F-CSE

The C2F-CSE module modifies a basic module of C2F by adding a consecutive selective enhancement (CSE) module. As illustrated in Figure 2, two attention modules are developed to improve the model's capability in capturing and utilizing spatial and channel information, respectively. The proposed module can increase the ability of extractor features to discriminate between vital information and trivial features. Its objective is to focus more on valuable features in the feature input. Besides, it pays attention to the critical context of the image object. The improvement also aims to enhance the model's ability to capture features of interest while optimizing the efficiency and performance of the model.

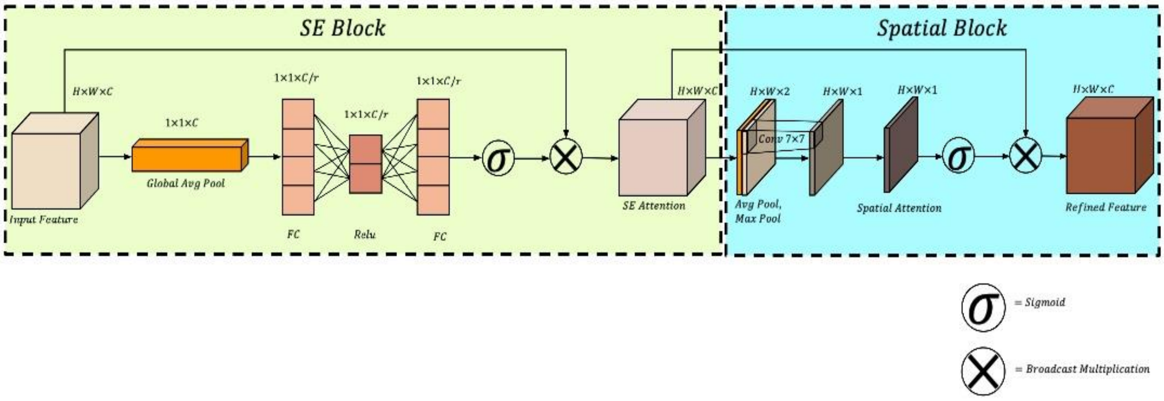


Figure 2. The proposed CSE module combines channel and spatial representation. It implements a squeeze and excitation module at the beginning of the module to highlight vital information along the channel map and spatial enhancement to capture the valuable features in a larger spatial area

2.1.3. CSE module

The proposed network utilizes the squeeze-and-excitation (SE) [23] and spatial attention blocks' feature-selective ability. This module is believed to improve the precision of the detection network by summarizing the representation feature and generating weighted scaling. A SE block is the first attention module designed to improve network performance by explicitly modeling the relationship between feature channels through a feature recalibration process. This process assigns weights to each feature channel. Feature recalibration applies a squeeze technique incorporating spatial information into the channel descriptors, and the excitation process learns the channel's activation corresponding to the input. This module can formulate as (1):

$$SE(X) = \sigma\left(W_2 ReLU(W_1 Z_C)\right) \otimes X, \tag{1}$$

where

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c}. \tag{2}$$

In the initial process, the input information (X) is modeled by capturing the global average region of the feature map across each channel through the operation of Z_C. The represented feature is then processed through two fully connected layers to model channel dependencies. This process is followed by applying the rectified linear unit (ReLU) function. This activation eliminates negative input values, thereby preventing irrelevant or detrimental information propagation in subsequent computations. It ensures that critical neurons are not hindered by low or negative scores, enabling the model to focus on valuable features effectively. The weights of the two fully connected layers are denoted as W_1 and W_2, and a sigmoid function (σ) is employed to generate weighted probability scores. Subsequently, the output vector from the sigmoid activation is multiplied with the original input feature map to refine the initial information based on channel-wise representation. The SE network provides only channel-specific attention in feature extraction and lacks enhanced spatial representation. Therefore, this study incorporates a spatial attention module to improve the enhancement of features. This addition allows the network to find interesting information in spatial coverage, enabling it to recognize specific dimension patterns to indicate fall features. The channel and spatial information combination can accurately understand unbalanced body positions as indications of a fall while ignoring irrelevant areas. In detail, the fusion module can illustrate as (3):

$$CSE(X) = \sigma(Conv^{7x7}[Avgpool(SE(X)), Maxpool(SE(X))]) \otimes SE(X). \tag{3}$$

The SE network boosts the quality of input features, which can improve performance by assigning more relevant weight to important feature channels. The output then adaptively reweights essential features in the spatial dimension, enhancing the model's performance on spatially mapped features. Average pooling (*AvgPool*) and max pooling (*MaxPool*) are used in parallel blocks to obtain feature summaries. The two spatial features are fused using the concatenate operation (*[]*), and then a 7×7 convolution filter ($Conv^{7x7}$)

extracts feature map information to cover a wider receptive field and detect more complex patterns. Finally, sigmoid activation (σ) emphasizes and highlights the spatial attention map. The proposed research integrates the modified SE network into the backbone structure of the YOLOv8 architecture to extract deeper features by emphasizing the learned weight represented in spatial and channel maps. The combined enhancement module improves the network's accuracy in recognizing and emphasizing essential features. The model also focuses on efficiently operating in a realistic application system. Moreover, the attention module can enhance the optimization of the feature learning process.

2.1.4. SPPF

A spatial pyramid pooling faster (SPPF) is an important component in the backbone that aims to increase object detection capability with high efficiency on diverse input features. This module optimizes the original version of spatial pyramid pooling (SPP) by pooling features using varying kernel sizes (*e.g.*, 5×5, 9×9, 13×13). Then, the results are combined to create a richer and diverse feature representation. This process helps the model capture information at different scales, making detecting objects of various sizes in images easier. Additionally, this module improves computational efficiency. It can speed up the inference process and strengthen the detection precision, making it an exciting component in the YOLOv8 architecture, especially in the backbone part.

2.2. Neck

The neck module aims to receive and combine features from various resolution levels produced in the backbone and then connect the information from the backbone to the head. It helps improve the feature representation before passing it to the final prediction. The PANet module is adopted to enable feature extraction from different levels of resolution on the map by enhancing the model to recognize different-sized information. PANet utilizes rapid feature fusion by extracting more comprehensive information. A light bottleneck module is offered in the C2F module at each prediction layer. This bottleneck structure variation enhances feature extraction effectiveness while reducing computational cost.

In order to improve the efficiency of the network, it proposes C2F-Next. This module is inspired by the basic module of convolutional two Faster, but the bottleneck part is modified using the light bottleneck. The structure of the light bottleneck applies the standard bottleneck design [26], as presented in Figure 3. The module can reduce computational cost while maintaining extraction ability without significantly declining precision. Depthwise convolution applies a single channel of filter operation that compromises mixed information of each channel input. This process can save many parameters and a rapid extraction process. The bottleneck module structure incorporates several block structures, which adopt depthwise operation (DW) at input information (X) using a 5×5 kernel, as shown in Figure 4. The large filter captures a sizable spatial area from input features and helps the network to increase the variety of the element object relationship. Furthermore, it utilizes LayerNorm (LN) to process each feature within a layer by normalizing each sample's information. A light bottleneck is formulated as (4):

$$LB(X) = PW\left(GR\big(SL(X)\big)\right), \tag{4}$$

where

$$SL(X) = PW\left(LN\big(DW(X)\big)\right). \tag{5}$$

This module applies pointwise convolution (*PW*), which employs a 1×1 convolutional block to integrate information from various channels while preserving the spatial dimensions of the features. The Gaussian error linear unit (GELU) activation function also helps optimize the model's performance by implementing a Gaussian approach to generate small negative inputs. Subsequently, global response normalization (GR) normalizes the activation output across all spatial features within a layer, thus enhancing the training process and improving model performance. Finally, pointwise convolution is applied in the last module to reconstruct the output channel and refine single spatial features, resulting in a more precise model. The proposed module focuses on efficiently extracting combination features in the neck stage. It utilizes light operation with a linear process without involving multi-channel mixing. It ensures that the selected feature works effectively, compromising the computational load.

2.3. Head

In its final stage, the model employs three heads that constitute a neural network responsible for predicting the locations and classes of objects. These heads determine the corresponding classes' bounding box locations and dimensions without adding additional computational overhead. These output heads are trained to generate offset scores attached to the final layers. The types of output heads can vary depending on the object detection algorithm and task requirements. Instead of using anchors for predicted boxes, YOLOv8 adopts an anchor-free detection method that directly predicts an object's center rather than its offset from a predefined box. This approach reduces the number of box predictions, speeds up post-processing, and simplifies the network, making it faster and improving the suitability of the proposed model with low-cost hardware configurations. However, object detection often encounters inaccuracies or missed detections, leading to errors. The intersection over union (IoU) metric addresses this by calculating the ratio of the overlap area of bounding boxes to their union area. It can further be used to compute the complete IoU (CIoU), which incorporates factors such as the distance between bounding box centers and an aspect ratio for scale. Distribution focal loss (DFL) and binary cross-entropy (BCE) loss functions are employed to evaluate the bounding boxes regression and classification accuracy of detected objects, respectively [27]. These critical loss functions play a pivotal role in training the model, enabling enhancements in predictive performance across successive iterations.

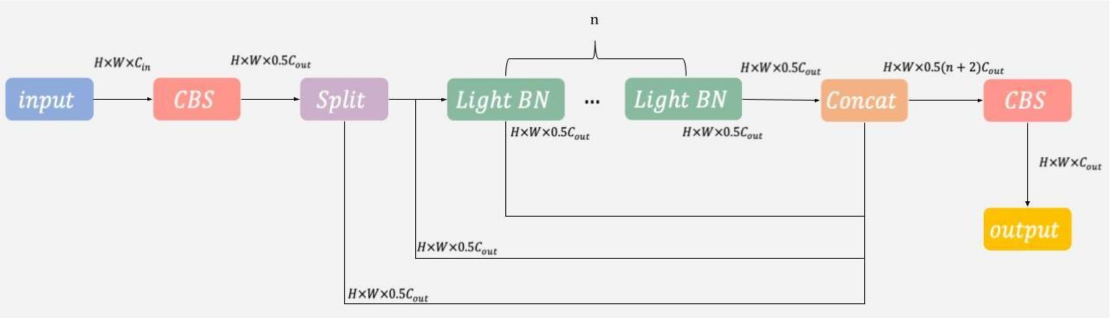

n
$H\times W\times C_{in}$
$H\times W\times 0.5C_{out}$
input
CBS
Split
Light BN
...
Light BN
Concat
CBS
output
$H\times W\times 0.5C_{out}$
$H\times W\times 0.5(n+2)C_{out}$
$H\times W\times 0.5C_{out}$
$H\times W\times 0.5C_{out}$
$H\times W\times C_{out}$
$H\times W\times 0.5C_{out}$
$H\times W\times 0.5C_{out}$

Figure 3. Modified C2f with light bottleneck applies efficient operation. It only provides an extensive computational effort on half of the parts but does not ignore the features of the rest

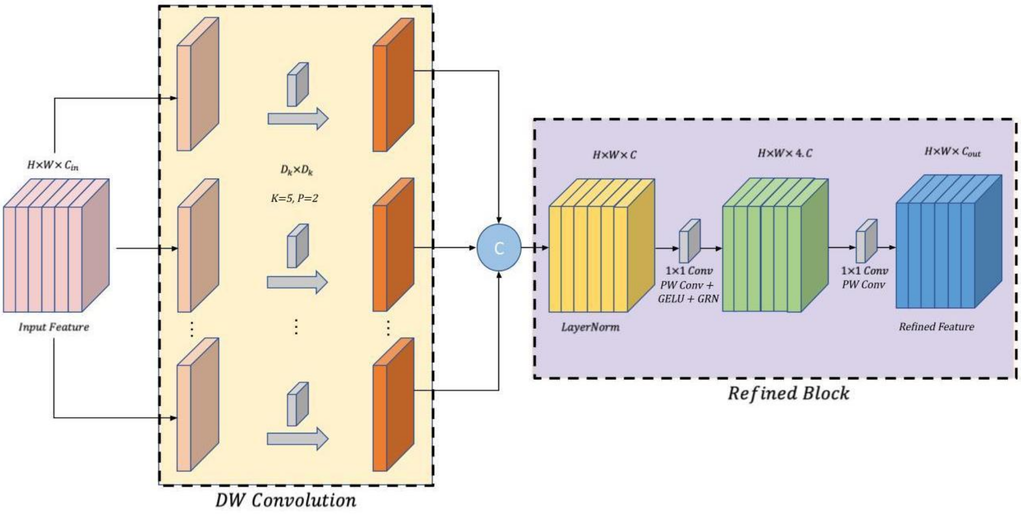


Figure 4. The proposed light bottleneck module applies a large kernel operation. It utilizes depthwise convolution with a large kernel size to capture a wider spatial area

3. DATASET AND IMPLEMENTATION SETUP

3.1. Dataset

The fall detection dataset uses Le2i [22] and consists of 2 classes: standing and falling human subjects. These images are sourced from the credible website Roboflow, which focuses on two main objectives: standing detection and falling detection. This dataset comprises 3010 images, divided into training, testing, and validation subsets, with a dataset split of 70% for training, 20% for testing, and 10% for validation. The second fallen dataset [28] consists of 3 classes: fallen, sitting, and standing. The dataset includes 3,290 images, divided into three parts: 74% training, 15% validation, and 10% testing.

3.2. Implementation setup

The proposed method utilizes the PyTorch framework and requires specialized hardware, including an AMD Ryzen 5 4500 6-core CPU @ 4.2 GHz process, 32 GB RAM, and an RTX 4060TI graphics card, to increase the training speed. The system utilizes the fall detection dataset for training and evaluation. The evaluation process relies on the values of average precision (AP) and mean average precision (mAP), using an IoU threshold of 0.5. The training was performed over 200 epochs with a batch size of 16. The optimizer used was Stochastic gradient descent (SGD), with a momentum of 0.937 and a learning rate of 0.01. This work implements mosaic augmentation that utilizes crop, flip, zoom, and shift geometry approaches to enrich the data challenges. The mosaic only was conducted on 190 epochs, and the remainder used normal mode. The image in the overall experiment is generalized with the input size dimension of 640×640. In testing speed, the model embedded in a Jetson Nano 4 GB representing an IoT device that directly connects with a webcam Eyesec 4K in live stream mode.

4. EXPERIMENT AND RESULTS

This section investigates the proposed model's evaluation results with two fall detection datasets that measure conformance to intersection over union. This experiment also compares the mean average precision (mAP) performance with other lightweight detection models within the scope of the YOLO family. In addition, efficiency comparisons are conducted by measuring the number of parameters, efficiency, and data processing speed. Furthermore, a comprehensive analysis of the model is presented in this study, which finds the usage impact of the proposed modules.

4.1. Ablation study

The ablation study presents the proposed module investigation that improves the performance of the YOLOv8 nano version. The intended network YOLOv8h-LB-CSE was compared at each step with modified block structures to see the impact of modifications. As shown in Table 1, the YOLOv8h-LB-CSE module into the original YOLOv8n has a notable decrease in Parameters by 60.34% and FLOP by 22.22%. In addition, there is an improvement in the performance model by 1.17% and 0.69% on the Le2i and Fallen datasets, respectively. The researchers also reconstructed the channel dimensions of the original YOLOV8 nano, which was limited to a maximum of 128 channels. This modified model is called YOLOV8h (Hypernano) because there is a significant decrease in learnable parameters by 69.84% and the number of operations by 26.39%. Moreover, it added a light bottleneck module to the YOLOV8-Hypernano module structure, which helps improve accuracy without significantly sacrificing computation cost. The light bottleneck module structure is designed to maintain efficiency and feature extraction capabilities, and the combined model is called YOLOv8h-LB.

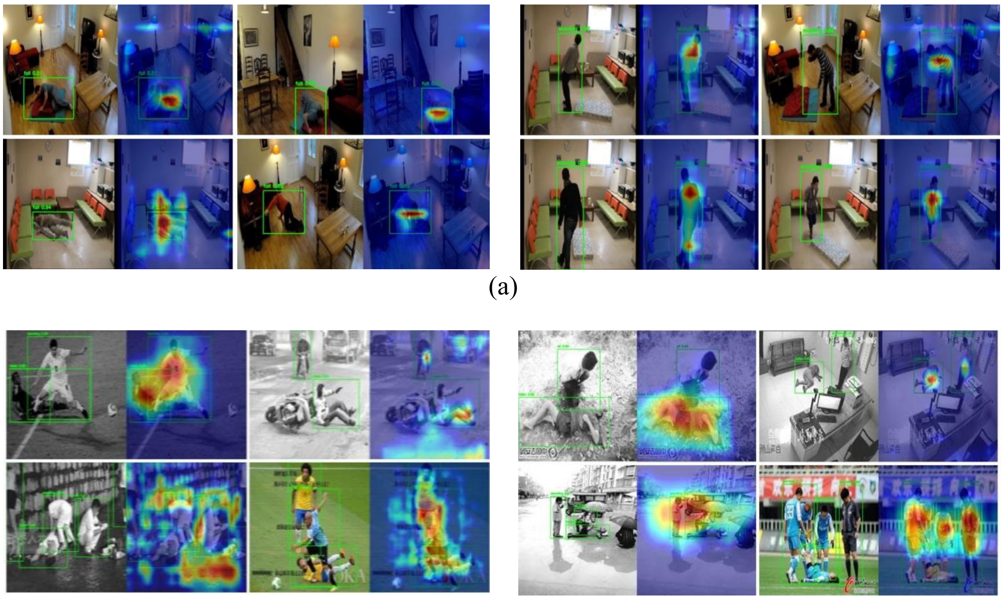
Table 1. Ablation experiments with different improvement strategies. It adds the proposed modules until they reach the entire proposed network

Models	GFLOPS	Parameters	mAP @0.5:0.95 on Le2i dataset	mAP @0.5 on Fallen dataset
YOLOv8n	7.2	3,011,238	0.596	0.727
YOLOv8h	5.3	907,238	0.573	0.708
YOLOv8h-LB	5.8	1,185,598	0.571	0.712
YOLOv8h-LB-CSE	**5.6**	**1,194,440**	**0.603**	**0.732**

Furthermore, the YOLOv8h with LB and CSE combines a light bottleneck with the proposed enhancement module, designed to improve network performance by recalibrating the input features. These findings prove that the enhanced YOLOv8 provides superior detection efficacy in fall detection. It also benefits from the lightweight incorporation of modules, leading to reduced model complexity. Including the light bottleneck and Squeeze-and-Excitation modules allows the model to capture essential information and recalibrate features effectively, improving overall accuracy without adding significant computational overhead. This enhancement encourages YOLOv8 to be particularly suitable for deployment in real-world applications with limited computational resources. Reducing the number of parameters and FLOPs makes the model more efficient and faster, which is crucial for real-time fall detection systems. Furthermore, carefully reconstructing the channel dimensions ensures that the model remains compact while maintaining high performance, making it an ideal solution for edge computing devices.

4.2. Evaluation on datasets

This study conducted a visual analysis to illustrate the detection performance of the modified YOLOv8 under various conditions, as shown in Figures 5(a) and 5(b). Each set of test images consists of two components: falling and standing categories. The left part presents the original photo, while the right part illustrates the heatmap results of the modified YOLOv8 algorithm. This visualization utilizes the Eigen-CAM approach, highlighting the most important features in red pixels. The heatmap of the falling category is shown in Figure 5(a), demonstrating that the proposed model emphasizes valuable information on the body part of the fallen object. As a result, this model can effectively recognize the falling position. For the standing category, the heatmap indicates that the model focuses on the correct prediction, showing that the heatmap area is inclined vertically. It also highlights the shoulders and feet as the main indicators of the standing position.



(a)

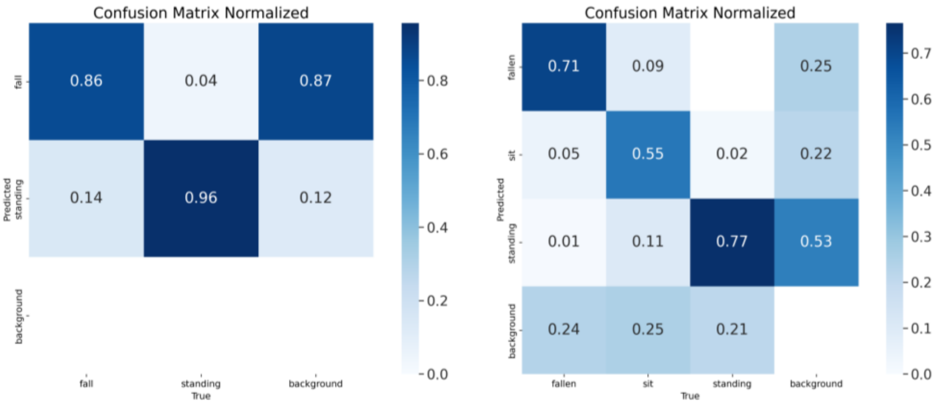
(b)

Figure 5. Heatmap observation of the proposed detector. It tests on (a) Le2i and (b) Fallen datasets. The target object is detected through green boxes

This study investigates the mean average precision of each prediction against each class label. The confusion matrix on the Le2i dataset is presented in Figure 6(a). The fall class exhibits the highest accuracy, with 0.96 instances correctly classified. However, there is a 0.04 misclassification rate, where instances of falls are incorrectly classified as "standing." Conversely, the standing class has a correct classification rate of 0.89 but obtains a 0.11 misclassification rate, with instances that should be classified as standing being incorrectly identified as fall. This analysis highlights the model's strengths and areas for improvement in distinguishing between fall and standing events.

Figure 6(b) illustrates the confusion matrix for the Fallen dataset, which includes three classes: fallen, sitting, and standing. The standing class exhibits the highest correct classification rate at 0.81. However, there are misclassification scores of 0.02 in standing instances, incorrectly classifying it as sitting, and background misclassification of 0.16. For the fallen class, the correct prediction rate is 0.66, but there are fallen misclassifications of 0.06 predicted as sitting and background misclassifications of 0.28. The sitting class obtains the lowest correct prediction rate of 0.60, with misclassifications of 0.07, 0.10, and 0.23 as fallen, standing, and background, respectively. This analysis indicates areas where the model's accuracy can improve, particularly distinguishing between similar postures.

This work compares the performance of the proposed model with the efficient YOLO families. It shows that our detector is superior to competitors, such as YOLOv3 tiny, YOLOv5n, YOLOv6n, YOLOv7 tiny, YOLOv8n, and YOLOv10n. Performance evaluation shows that YOLOv8h-LB-CSE achieves the best precision, measured with mAP of 0.603 in 0.5:95 IoU. The proposed model outperforms the original YOLOv8n, differing by a mAP of 5.61%. Even it is superior to the new version of the lightweight YOLOv10n. This observation also compares the performance of YOLOv8h-LB with several attention modules, and the majority presents improving precision. An enhancement block is assigned to increase the ability of extraction features, although it only does not significantly add to the computation cost. The proposed model with CSE obtains a mAP of 5.98% higher than YOLOv8h-LB-SE, which uses the Squeeze Excitation attention module. This structure only focuses on channel-wise enhancement. Furthermore, the proposed model also performs better than CBAM attention, which differs by 2% mAP. CBAM uses a configuration similar to CSE but with over-in-channel extraction. Compared to the CAN and ELA attention modules, our network shows higher mAP. Although both of these attentions involve context discovery from the spatial regions of the map, they are not robust enough to discriminate between falling and standing features. GCNET and DAN performed well in the object detection task but could not outperform CSE.



(a) (b)

Figure 6. Confusion matrices of model prediction. It evaluates on (a) Le2i and (b) Fallen datasets

On the other hand, Table 2 shows a comparison of YOLOv8h-LB-CSE performance with other detectors on the Fallen dataset. The proposed network achieves mAP@0.5 of 0.732, indicating that our model outperforms YOLOv8n by 0.688% mAP and surpasses YOLOv7 tiny by 3.39%. On this dataset, the proposed model also compares the precision with other YOLO lightweight models. Although YOLOv6n outperforms our detector, other lightweight detectors underperform. YOLOv6n achieves higher precision than the proposed model by 0.003, but the model generates more parameters and computation. Moreover, this satisfactory performance also outperforms the mAP of YOLOv5n by 0.021. A comparison with the state-of-the-art network, YOLOv10n, shows that our detector is superior by 0.2%. The Hypernano version shows a lower performance than the full proposed network. This result represents that the proposed gain module can improve performance in recognizing falling, sitting, and standing activities, thereby demonstrating the practical implications of our work.

4.3. Evaluation of model efficiency

The design of the proposed model considers the advantages that help the application scenario. The proposed model is designed with attention to several important aspects, such as a low number of giga floating-point operations per seconds (GFLOPS) and a minimal number of parameters compared to other models. This research prioritizes efficiency; a more efficient model can perform many computational tasks with fewer operations and fewer trainable weights. This issue is directly related to the number of GFLOPS and the total number of parameters. Analysis of the comparative experiments YOLOv8h-LB-CSE is the cheapest model. The proposed model is very lightweight compared to the efficient YOLO detectors, as shown in Tables 2 and 3. The original version of YOLOv8n generates 2.5 times larger than our proposed detector. Our detector uses only 1.46 times less operation usage. Furthermore, the parameters and number of operations are widely used by YOLOv3 tiny and YOLOv7 tiny. Thus, this also requires a significant processing device memory while weakening the data processing speed.

The speed of data processing determines the reliability of a method when implemented on a device. In order to support the smart IoT system, this work evaluates the proposed model speed on an NVIDIA Jetson Nano with a VRAM of 4 GB. This device is commonly used as an edge device for intelligent systems. Based on the graph in Figure 7, the proposed model with light bottleneck and CSE achieved a speed of 10.64 FPS when tested on the device. Compared to the lightweight model YOLOv8h, it is reduced by 25.4%. On the other hand, YOLOv8h-LB-CSE achieves 40.1% faster compared to YOLOv3-tiny. Another comparison is that the proposed model is 8.5% and 9.8% slower than YOLOv8n and YOLOv6n, respectively.

A comparison with the most popular YOLO versions, such as the YOLOv5n, shows that our model is slower than this competitor's, which is our model's weakness. The efficiency of the number of parameters and computational complexity of the proposed model outperforms that of the efficient YOLO detectors. However, it requires more processing memory. It is due to depth-wise operations that apply branching operations for each channel. Hence, this operation requires more memory than the regular convolution operation. On the other hand, the speed achieved by our detector of 10.64 FPS is feasible for smooth operation on edge devices that support IoT intelligent systems. The priority emphasizes fall detection performance that minimizes fall activity recognition errors.

Table 2. Comparison of the proposed detector with other lightweight YOLO detectors and attention mechanism methods on the Fallen dataset. It also compares the number of parameters and computational complexity of models

Models	GFLOPS	Parameters	mAP @0.5	mAP @0.5:0.95
YOLOv3 tiny	19.0	12,133,670	0.68	0.365
YOLOv5n	7.2	2,509,049	0.711	0.387
YOLOv6n	11.9	4,238,441	0.735	0.410
YOLOv7 tiny	13.2	6.020.400	0,690	0,320
YOLOv8n	8.2	3,011,433	0,727	0,405
YOLOv10n	6.2	2,695,196	0.711	0.396
YOLOV8h	5.3	907,433	0.708	0.387
YOLOv8h-LB	5.4	917.417	0.712	0.387
YOLOv8h-LB-CSE	**5.6**	**1.194.635**	**0.732**	**0.398**

Table 3. Comparison of the proposed detector with other lightweight YOLO detectors and attention mechanism methods on the Le2i dataset. It also compares the number of parameters and computational complexity of models

Models	GFLOPS	Parameters	mAP @0.5	mAP @0.5:0.95
YOLOv3 tiny	19.0	12,133,156	0.85	0.576
YOLOv5n	7.2	2,508,854	0.889	0.584
YOLOv6n	11.9	4,238,342	0.859	0.549
YOLOv7 tiny	13.2	6,017,694	0.898	0.532
YOLOv8n	8.2	3,011,238	0,910	0.571
YOLOv8h	5.3	907,238	0,907	0.574
YOLOv10n	6.2	2,695,196	0.893	0.525
YOLOv8h-LB	5.4	917,222	0,907	0.577
YOLOv8h-LB-SE	5.8	1,182,600	0.918	0.569
YOLOv8h-LB-CBAM	5.8	1,184,392	0.913	0.583
YOLOv8h-LB-CAN	5.8	1,185,598	0.89	0.571
YOLOv8h-LB-ELA	5.8	1,184,550	0.906	0.594
YOLOv8h-LB-DAN	5.8	1,219,400	0.909	0.577
YOLOv8-h-LB-GCNET	5.8	1,184,575	0.898	0.571
YOLOv8h-LB-ECA	5.8	1,182,249	0.876	0.57
YOLOv8h-LB-CSE	**5.6**	**1.194.440**	**0,916**	**0,603**

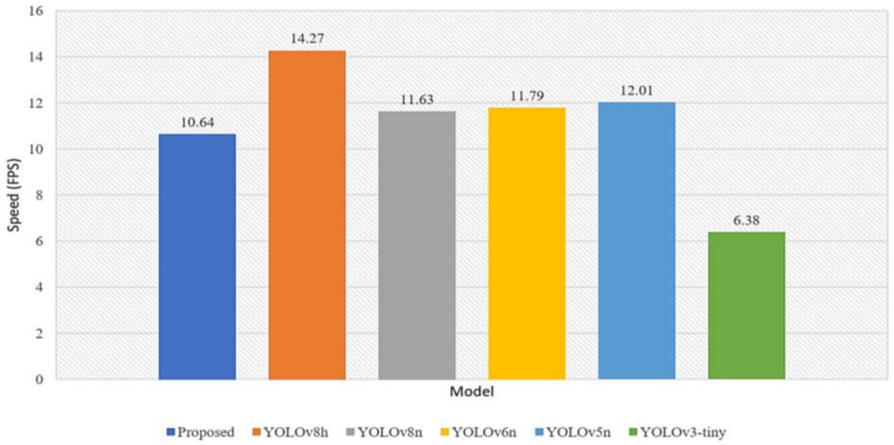


Figure 7. Speed comparison of the proposed model with other lightweight YOLO models. The proposed model achieves 10.64 FPS faster than YOLOv3-tiny

4.3. Practical scenario testing and future research

Practical applications demand that vision algorithms operate efficiently on embedded devices and deliver high accuracy. Our model was tested in a real-world scenario to evaluate the reliability of our proposed system and implemented as an IoT-based intelligent system for monitoring falls. Live video streams were captured from an RGB webcam and processed on a Jetson Nano, which served as the computational platform for running the model. The IoT setup was installed on a high wall corner, simulating an intelligent video surveillance environment. It was trained using the Fallen dataset to ensure the model's effectiveness in real-world detection, which classifies actions into three categories: falling, sitting, and standing. The results demonstrate that the proposed fall detection system operates efficiently, accurately recognizing these actions, as illustrated in Figure 8. The visualizations reveal that our system effectively detects falls, indicated by red bounding boxes around the falling individual. The system demonstrates high accuracy in fall detection and correctly predicting sitting and standing. However, during the movement transition process, the model occasionally misclassifies actions. Figure 8 (bottom row) presents some of these prediction errors. It causes the limited variety of motion data for transition phases. Additionally, the model requires more temporal awareness, as it needs to account for the sequential relationships in time-series data, further contributing to these scenes' inaccuracies.

Our work introduces a lightweight CNN architecture that generates low trainable parameters and reduces computational overhead. However, using depthwise convolutions impacts data processing speed due to branching operations, which increase memory usage. Future work will optimize these branching operations to improve the speed system. One potential approach is reducing reliance on depthwise convolutions by incorporating grouped convolution operations, which can alleviate memory bottlenecks. Improving data processing speed will enhance the model's applicability in real-world scenarios, particularly optimizing fall detection systems. Additionally, future work will address high-frequency features by incorporating attention modules and revisiting error functions, improving detection performance and reducing misclassifications related to fall events. Strengthening the relationship between features will also mitigate the loss of critical features caused by excessive convolution operations, further boosting the model's accuracy and robustness.

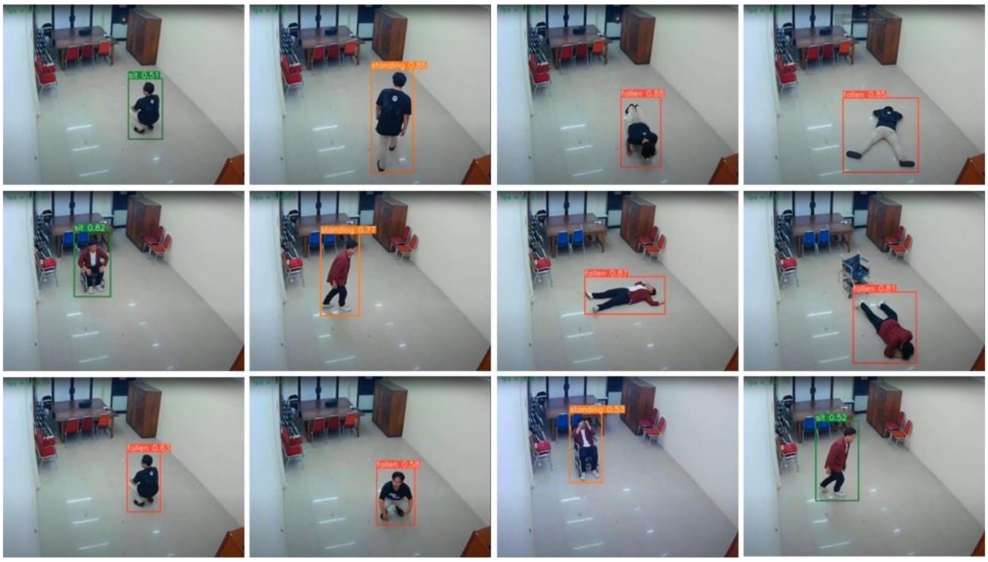


Figure 8. Visualization of fall detection results in real case scenarios. These scenes were performed in a laboratory environment

5. CONCLUSION

This research introduces a lightweight network that improves YOLOv8n, a new efficient method for human fall detection designed to address the computational challenges of using conventional YOLOv8 in fall detection scenarios. The study presents YOLOv8-Hypernano (YOLOv8h), which enhances the model's efficiency and performance by reducing the number of channels and keeping the performance. It combines spatial and channel attention modules in the backbone, improving the focus on human subjects by more accurately detecting motion patterns. It installs a consecutive selective enhancement (CSE) module to improve the efficiency and effectiveness of feature extraction while reducing computational costs. The neck structure is also modified with a lightweight bottleneck network that cautiously reconstructs feature maps at depth layers. It avoids abundant operations to accurate human motion patterns and maintains efficiency in feature extraction. Moreover, YOLOv8-Hypernano with CSE outperforms other advanced lightweight algorithms such as YOLOv3-tiny, YOLOv5-nano, YOLOv6-nano, YOLOv7-tiny, and YOLOv8-nano. The model evaluation results show that the proposed detector achieves a mAP score of 0.603 and 0.732 on the Fallen and Le2i datasets. The model generates parameters of 1,194,440 and computations of 5.6 G. Visualization results indicate that the proposed detector works optimally and is suitable for implementation in an IoT system. Further work is required to improve the performance of the detector head and refine the high-level features.

FUNDING INFORMATION

This research is funded by Fundamental Research of UNSRAT Excellence Cluster 1 (RDUU K1) Sam Ratulangi University 2024 with contract number 192/UN12.27/LT/2024.

REFERENCES

[1] K. L. Lu and E. T. H. Chu, “An image-based fall detection system for the elderly,” *Applied Sciences (Switzerland)*, vol. 8, no. 10, 2018, doi: 10.3390/app8101995.

[2] B. H. Wang, J. Yu, K. Wang, X. Y. Bao, and K. M. Mao, “Fall detection based on dual-channel feature integration,” *IEEE Access*, vol. 8, pp. 103443–103453, 2020, doi: 10.1109/ACCESS.2020.2999503.

[3] Z. Qian *et al.*, “Development of a real-time wearable fall detection system in the context of internet of things,” *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21999–22007, 2022, doi: 10.1109/JIOT.2022.3181701.

[4] N. Waleed and M. S. Jarjees, “IoT based vital signs monitoring with fall detection system,” *International Conference on Engineering, Science and Advanced Technology, ICESAT 2023*, pp. 81–85, 2023, doi: 10.1109/ICESAT58213.2023.10347289.

[5] D. Mohan *et al.*, “Artificial intelligence and IoT in elderly fall prevention: a review,” *IEEE Sensors Journal*, vol. 24, no. 4, pp. 4181–4198, 2024, doi: 10.1109/JSEN.2023.3344605.

[6] S. Phatangare, S. Kate, D. Khandelwal, A. Khandetod, and A. Kharade, “Real time human activity detection using YOLOv7,” *7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2023 - Proceedings*, pp. 1069–1076, 2023, doi: 10.1109/I-SMAC58438.2023.10290168.

[7] T. Chen, Z. Ding, and B. Li, “Elderly fall detection based on improved YOLOv5s network,” *IEEE Access*, vol. 10, pp. 91273–91282, 2022, doi: 10.1109/ACCESS.2022.3202293.

[8] A. Raza, M. H. Yousaf, and S. A. Velastin, “Human fall detection using YOLO: a real-time and AI-on-the-edge perspective,” *2022 12th International Conference on Pattern Recognition Systems, ICPRS 2022*, 2022, doi: 10.1109/ICPRS54038.2022.9854070.

[9] Y. Thwe, N. Jongsawat, and A. Tungkasthan, “Accurate fashion and accessories detection for mobile application based on deep learning,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 4, pp. 4347–4356, 2023, doi: 10.11591/ijece.v13i4.pp4347-4356.

[10] N. R. Kolukula, R. P. Kalapala, S. S. R. Ivaturi, R. K. Tammineni, M. Annavarapu, and U. Pyla, “An efficient object detection by autonomous vehicle using deep learning,” *International Journal of Electrical and Computer Engineering*, vol. 14, no. 4, pp. 4287–4295, 2024, doi: 10.11591/ijece.v14i4.pp4287-4295.

[11] E. A. Mahareek, E. K. Elsayed, N. M. ElDesouky, and K. A. Eldahshan, “Detecting anomalies in security cameras with 3D-convolutional neural network and convolutional long short-term memory,” *International Journal of Electrical and Computer Engineering*, vol. 14, no. 1, pp. 993–1004, 2024, doi: 10.11591/ijece.v14i1.pp993-1004.

[12] W. Toghuj and Y. Alraba’nah, “A two-stage approach for aircraft detection with convolutional neural network,” *International Journal of Electrical and Computer Engineering*, vol. 14, no. 4, pp. 4627–4635, 2024, doi: 10.11591/ijece.v14i4.pp4627-4635.

[13] M. D. Putro, D. L. Nguyen, and K. H. Jo, “A fast CPU real-time facial expression detector using sequential attention network for human–robot interaction,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7665–7674, 2022, doi: 10.1109/TII.2022.3145862.

[14] M. D. Putro, J. Litouw, and V. C. Poekoel, “Low-resolution facial emotion recognition on low-cost devices,” *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 2199–2209, 2024, doi: 10.11591/ijai.v13.i2.pp2201-2211.

[15] Y. Yin, L. Lei, M. Liang, X. Li, Y. He, and L. Qin, “Research on fall detection algorithm for the elderly living alone based on YOLO,” in *Proceedings of 2021 IEEE International Conference on Emergency Science and Information Technology, ICESIT 2021*, 2021, pp. 403–408, doi: 10.1109/ICESIT53460.2021.9696459.

[16] J. Gutiérrez, V. Rodríguez, and S. Martin, “Comprehensive review of vision-based fall detection systems,” *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–50, 2021, doi: 10.3390/s21030947.

[17] H. Wang, C. Liu, Y. Cai, L. Chen, and Y. Li, “YOLOv8-QSD: an improved small object detection algorithm for autonomous vehicles based on YOLOv8,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–16, 2024, doi: 10.1109/TIM.2024.3379090.

[18] H. Yi, B. Liu, B. Zhao, and E. Liu, “Small object detection algorithm based on improved YOLOv8 for remote sensing,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 1734–1747, 2024, doi: 10.1109/JSTARS.2023.3339235.

[19] Y. Dai and W. Liu, “GL-YOLO-Lite: a novel lightweight fallen person detection model,” *Entropy*, vol. 25, no. 4, 2023, doi: 10.3390/e25040587.

[20] G. Shen, B. Zhao, X. Chen, L. Liu, Y. Wei, and T. Yin, “Human fall detection based on re-parameterization and feature enhancement,” *IEEE Access*, vol. 11, pp. 133591–133606, 2023, doi: 10.1109/ACCESS.2023.3335833.

[21] Y. Wang, Z. Chi, M. Liu, G. Li, and S. Ding, “High-performance lightweight fall detection with an improved YOLOv5s algorithm,” *Machines*, vol. 11, no. 8, 2023, doi: 10.3390/machines11080818.

[22] X. Kan, S. Zhu, Y. Zhang, and C. Qian, “A lightweight human fall detection network,” *Sensors*, vol. 23, no. 22, 2023, doi: 10.3390/s23229069.

[23] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020, doi: 10.1109/TPAMI.2019.2913372.

[24] Q. Hou, D. Zhou, and J. Feng, “Coordinate attention for efficient mobile network design,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13708–13717, doi: 10.1109/CVPR46437.2021.01350.

[25] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, “EPSANet: an efficient pyramid squeeze attention block on convolutional neural network,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13843 LNCS, pp. 541–557, 2023, doi: 10.1007/978-3-031-26313-2_33.

[26] C. Li, C. Chen, Y. Hei, J. Mou, and W. Li, “An efficient advanced-YOLOv8 framework for THz object detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024, doi: 10.1109/TIM.2024.3394487.

[27] Z. J. Khow, Y. F. Tan, H. A. Karim, and H. A. A. Rashid, “Improved YOLOv8 model for a comprehensive approach to object detection and distance estimation,” *IEEE Access*, vol. 12, pp. 63754–63767, 2024, doi: 10.1109/ACCESS.2024.3396224.

[28] ByoungWook, “fallen_new_version Dataset,” *Roboflow Universe*. 2023.

BIOGRAPHIES OF AUTHORS

Pinrolinvic D. K. Manembu received a bachelor’s of engineering (S.T.) in electrical engineering from Sam Ratulangi University in Manado, Indonesia, in 2005. He received a Master of Science (MT) degree from the Faculty of Industrial Technology at Bandung Institute Technology, Indonesia, in 2012. In early 2013, he joined the Department of Electrical Engineering, at Sam Ratulangi University, as an assistant professor. His current research interests include artificial intelligence, the internet of things for energy efficiency, and robotics. He can be contacted at email: pmanembu@unsrat.ac.id.

Jane Ivonne Litouw received a Bachelor’s of Engineering (S.T.) in electrical engineering from Sam Ratulangi University in Manado, Indonesia, in 2003. He received a Magister Teknik (M.T) degree from STEI ITB in Bandung, Indonesia, in 2014. In 2005 she joined the Department of Electrical Engineering, Sam Ratulangi University, as lecturer. Her current research interests are fuzzy logic system, image processing and deep learning. She can be contacted by email: jane litouw@unsrat.ac.id.

Feisy Diane Kambey received a Bachelor’s of Engineering (S.T.) in electrical engineering at Sam Ratulangi University (UNSRAT) from 2000 to 2005. She then pursued a master's degree in control engineering and artificial intelligence at the Bandung Institute of Technology (ITB), graduating in 2013. Her research interests in control engineering, artificial intelligence, and machine learning. She can be reached via email at feisykambey@unsrat.ac.id.

Abdul Haris Junus Ontowirjo received the B.Eng. degree in electrical engineering from the Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 1991, M.Eng. degree in electrical engineering from the Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2010. His current research interest includes automation and intelligent control. He can be contacted at email: aharisjo@unsrat.ac.id.

Vecky Canisius Poekoel received a bachelor’s of engineering (S.T.) in electrical engineering from Institut Teknologi Sepuluh November (ITS) in Surabaya, Indonesia, in 1994. He received an M.T. degree from the Department of Electrical Engineering at Institut Teknologi Bandung (ITB), in Bandung, Indonesia, in 2005. He graduated a Dr.Eng. with the Department of Computer Science and Electrical Engineering, Kumamoto University, Kumamoto, Japan, in 2014. In 1994, he joined the Department of Electrical Engineering, Sam Ratulangi University. His current research interests include control engineering and artificial intelligence. He can be contacted at email: vecky.poekoel@unsrat.ac.id.

Muhamad Dwisnanto Putro received a bachelor’s of engineering (S.T.) in electrical engineering from Sam Ratulangi University in Manado, Indonesia, in 2010. He received an M.Eng. degree from the Department of Electrical Engineering at Gadjah Mada University in Yogyakarta, Indonesia, in 2012. He graduated a Ph.D. degree with the Department of Electrical, Electronic, and Computer Engineering, University of Ulsan, South Korea, in 2022. In 2013, he joined the Department of Electrical Engineering, Sam Ratulangi University, as an assistant professor. His current research interests include computer vision and deep learning, which focuses on robotic vision and perception. He can be contacted at email: dwisnantoputro@unsrat.ac.id.