Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.066343

ARTICLE

Enhancing Classroom Behavior Recognition with Lightweight Multi-Scale Feature Fusion

Chuanchuan Wang[1,2], **Ahmad Sufril Azlan Mohamed**[2,*], **Xiao Yang**[2], **Hao Zhang**[2], **Xiang Li**[1] **and Mohd Halim Bin Mohd Noor**[2]

[1]School of Engineering, Guangzhou College of Technology and Business, Guangzhou, 510850, China
[2]School of Computer Sciences, Universiti Sains Malaysia, Minden, 11800, Penang, Malaysia

*Corresponding Author: Ahmad Sufril Azlan Mohamed. Email: sufril@usm.my

Received: 05 April 2025; Accepted: 19 June 2025; Published: 29 August 2025

ABSTRACT: Classroom behavior recognition is a hot research topic, which plays a vital role in assessing and improving the quality of classroom teaching. However, existing classroom behavior recognition methods have challenges for high recognition accuracy with datasets with problems such as scenes with blurred pictures, and inconsistent objects. To address this challenge, we proposed an effective, lightweight object detector method called the RFNet model (YOLO-FR). The YOLO-FR is a lightweight and effective model. Specifically, for efficient multi-scale feature extraction, effective feature pyramid shared convolutional (FPSC) was designed to improve the feature extract performance by leveraging convolutional layers with varying dilation rates from the input image in the backbone. Secondly, to address the problem of multi-scale variability in the scene, we design the Rep Ghost fusion Cross Stage Partial and Efficient Layer Aggregation Network (RGCSPELAN) to improve the network performance further and reduce the amount of computation and the number of parameters. In addition, by conducting experimental valuation on the SCB dataset3 and STBD-08 dataset. Experimental results indicate that, compared to the baseline model, the RFNet model has increased mean accuracy precision (mAP@50) from 69.6% to 71.0% on the SCB dataset3 and from 91.8% to 93.1% on the STBD-08 dataset. The RFNet approach has effectiveness precision at 68.6%, surpassing the baseline method (YOLOv11) at 3.3% and archieve the minimal size (4.9 M) on the SCB dataset3. Finally, comparing it with other algorithms, it accurately detects student behavior in complex classroom environments results confirmed that RFNet is well-suited for real-time and efficiently recognizing classroom behaviors.

KEYWORDS: Classroom action recognition; YOLO-FR; feature pyramid shared convolutional; rep ghost cross stage partial efficient layer aggregation network (RGCSPELAN)

1 Introduction

In traditional classrooms, teachers can monitor only a small subset of students, hindering a comprehensive understanding of classroom dynamics [1]. Moreover, in crowded classrooms, it becomes difficult for teachers to monitor each student's learning progress and level of engagement effectively [2]. To solve these problems, real-time and high-accuracy classroom action recognition methods are essential, enabling teachers to dynamically adjust their teaching strategies and enhance student status for learning in the classroom.

The advent of smart classrooms and advances in deep learning technologies have paved the way for developing real-time classroom action recognition methods, which hold the potential to establish

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

closed-loop teaching environments and more efficient pedagogical practices [1]. Recently, a growing interest in leveraging computer vision techniques to recognize student actions and behaviors in classroom environments [3–5]. However, to safeguard student privacy, the majority of research datasets remain unpublished or inaccessible [6,7]. Those available are predominantly sourced from primary and secondary school classrooms or open online classroom videos, which fail to accurately capture the behavioral nuances of university students in real-world settings. Existing datasets span diverse subjects, classrooms, and student actions, encompassing behaviors such as listening, discussing, reading, writing, raising hands, turning around, and standing. In university classrooms, the complexity of student behavior is further heightened by crowded seating arrangements and frequent mutual occlusion among students, making real-time recognition of classroom actions a challenging task.

Monitoring systems oversee hundreds of classrooms, requiring real-time, multi-person action recognition [8]. Conducting classroom engagement analysis demands a model that is both lightweight and highly accurate. YOLOv11 [9], the latest iteration of the YOLO series, offers a compelling solution with its superior speed, precision, and low parameter count, making it one of the most advanced object detection frameworks.

This work uses YOLOv11 [9], with the fewest parameters and the fastest inference speed as the baseline network. To further optimize performance, we propose a feature pyramid shared convolutional (FPSC) to reduce the model's parameter count while enhancing its accuracy. Besides, we propose the Rep Ghost fusion Cross Stage Partial and Efficient Layer Aggregation Network (RGCSPELAN) to further improve the network performance and reduce the computation and parameters. Experimental results demonstrate that the improved YOLOv11 model delivers efficient performance in addressing the challenges of classroom action recognition.

The main contributions and innovations are: 1. To improve the feature extract performance for classroom action recognition, efficient multi-scale feature extraction, and the design of an effective feature pyramid shared convolutional. 2. To address the problem of multi-scale variability in the scene, we propose the Rep Ghost CSPELAN (RGCSPELAN) to improve the network performance further and reduce the amount of computation and the number of parameters. 3. The YOLO-RF model was evaluated on the SCB dataset3 and STBD-08 dataset, extensive experimental results demonstrate the effectiveness and robustness of the proposed algorithm on the two public datasets.

2 Related Work

2.1 Traditional Classroom Action Recognition Methods

A fast object detection method based on FFmpeg was proposed in [10] to enhance classroom behavior image recognition. This method extracts features from manually labeled target regions using a combination of Histogram of Oriented Gradients (HOG) and Motion History Images (MHI). A behavior recognition model is then constructed using a hybrid backpropagation neural network and support vector machine (BP-SVM) classifier, employing a lookup table to facilitate end-to-end feature extraction and automatic learning. The primary advantage of this approach lies in its ability to uncover underlying patterns and complex relationships within the data without the need for manual feature design and selection.

In contrast, convolutional neural networks (CNNs) adapt to data complexity and variability, often surpassing traditional methods on large-scale datasets with hierarchical feature extraction. For instance, a teacher behavior detection framework proposed in [11] extracts RGB image information and combines it with skeletal data to construct behavioral features, subsequently classifying actions using SVM. However, these earlier approaches rely heavily on handcrafted feature extraction, limiting their adaptability to more complex and diverse datasets.

To overcome these limitations, this study leverages CNNs for automatic feature learning, enabling the model to better handle the complexity and variability of classroom behavior data and achieve improved performance, particularly on larger datasets.

2.2 Deep Learning-Based Classroom Action Recognition Methods

2.2.1 Pose-Based Classroom Action Recognition Methods

Many studies have explored the use of convolutional neural networks (CNNs) to analyze human skeletal information for classroom behavior recognition [12]. Human pose estimation techniques leverage object detection frameworks, such as You Only Look Once (YOLO) [13] and Single Shot Detector (SSD) [14], to automatically detect and extract skeletal data. Behavior classification is then performed based on key points derived from this skeletal information. For example, Hur and Bosch [15] combined OpenPose with CNNs to analyze classroom video data, addressing challenges such as occlusions during behavior recognition.

While these approaches demonstrate promising results, detecting complete skeletal information in real-world classroom scenarios remains difficult due to frequent occlusions, which significantly affect detection accuracy.

2.2.2 CNN-Based Classroom Action Recognition Methods

Object detection methods have garnered significant attention in research due to their ability to simultaneously predict bounding boxes and classify behaviors, making them particularly suitable for real-time classroom behavior recognition. However, traditional object detection approaches often encounter challenges in classroom settings, including severe occlusion, low-resolution images, and the prevalence of numerous small objects. To address these limitations, Tang et al. [16] enhanced Faster R-CNN [3,17] by integrating merged ROI pooling and locality-preserving learning to boost detection performance. This design enables the network to retain richer semantic and high-resolution features, thereby enhancing classification accuracy.

Building on this foundation, subsequent improvements introduced a multiscale feature enrichment branch and an adaptive fusion mechanism [3], which increased the robustness and discriminative capability of the feature extractor. The model further employed adaptive smooth L1 loss during training, achieving superior performance compared to other methods when evaluated on classroom behavior datasets.

In addition, attention mechanisms, inspired by the human ability to filter relevant information, have been incorporated into CNN architectures to focus on the most critical features of input data, thus improving classroom behavior recognition. For instance, Wang et al. [18] utilized the squeeze-and-excitation (SE) attention mechanism, while Zhu et al. [19] implemented the convolutional block attention module (CBAM) in their CNN models. Both approaches enabled the networks to emphasize crucial regions within classroom behavior images, significantly enhancing recognition performance.

2.3 YOLOv11 Network

The YOLOv11 (You Only Look Once, version 11) [9], framework represents a state-of-the-art deep learning architecture for real-time object detection. It builds upon the foundations of previous YOLO versions, introducing advanced techniques to enhance detection accuracy, computational efficiency, and adaptability to diverse environments. Key innovations in YOLOv11 include a more efficient backbone for feature extraction, the integration of Transformer-based modules for improved global context understanding, and an optimized anchor-free mechanism to handle objects of varying scales. Additionally, it employs a lightweight pyramid structure to better capture multi-scale features, making it particularly suitable for

scenarios requiring high-speed processing and resource efficiency, such as edge computing and embedded systems. The overview architecture as shown in Fig. 1.

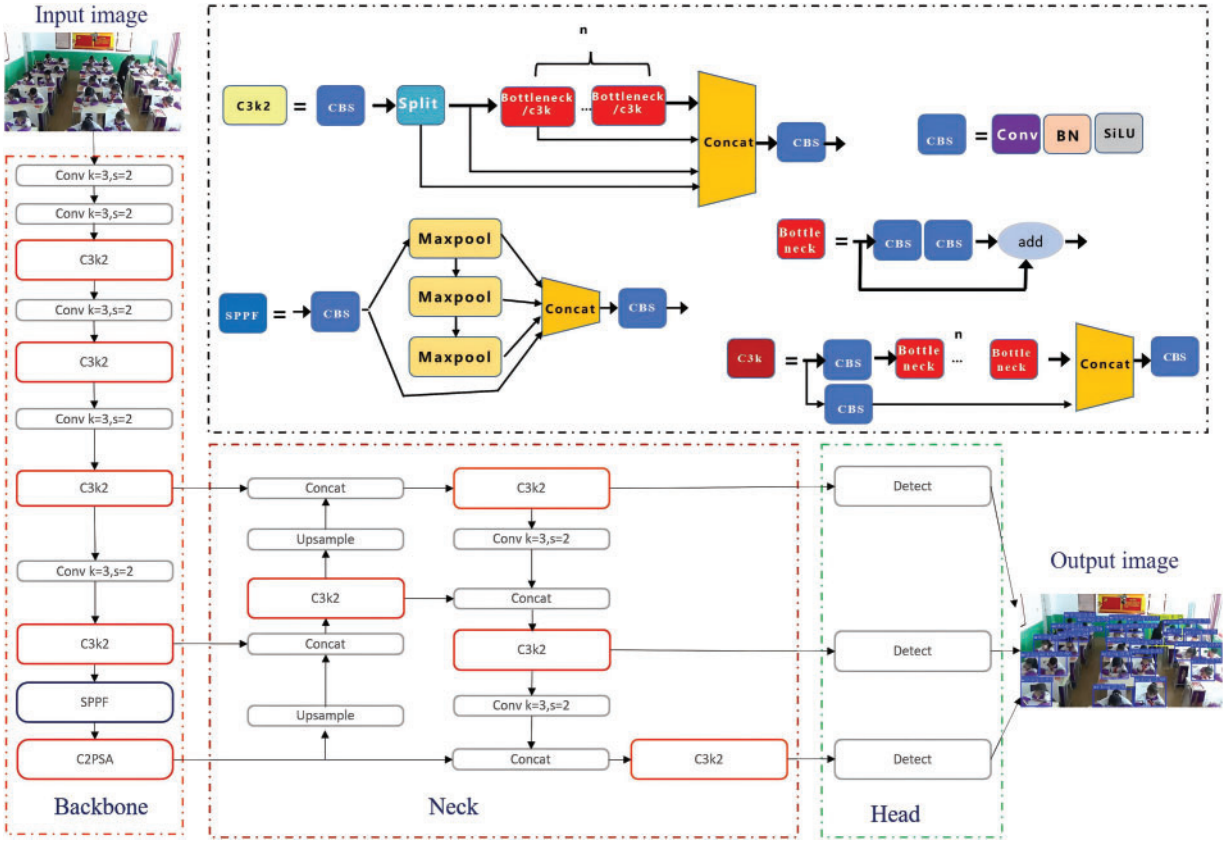


Figure 1: The YOLOv11 architecture

Compared to the YOLOv6 [20] and YOLOv8 [21] model, it changes the C2F module to C3K2, and also adds a C2PSA module behind the SPPF module, and introduces the YOLOv10 [22] HEAD idea into the YOLO11 HEAD, which uses the depth separable method to reduce redundant calculations and improve efficiency. The loss calculation and network structure enhancements represent significant progress in YOLOv11 [9]. However, it still suffers from the problems of a large number of computational parameters and a long time-consuming time, to solve these problems, the lightweight YOLO architecture is the focus of this work. Especially, in classroom behavior recognition scenarios, an effective lightweight object detector method seems necessary and urgent.

2.4 Summary

This section reviews the evolution of classroom action recognition methods, highlighting the transition from traditional handcrafted feature-based approaches (e.g., HOG-MHI with BP-SVM) to deep learning models utilizing convolutional neural networks (CNNs) and pose estimation. While CNNs and skeleton-based methods have improved recognition performance, challenges such as occlusion and low-resolution remain. Recent works have explored enhanced object detectors like Faster R-CNN [17] with attention mechanisms and feature fusion strategies to boost robustness. The study further introduces YOLOv11, a state-of-the-art detector with Transformer modules and anchor-free design, emphasizing its improvements

in accuracy, speed, and adaptability. However, due to high computational demands, lightweight adaptations of YOLOv11 are proposed for real-time classroom behavior recognition on resource-constrained devices.

3 Proposed Method

Existing classroom behavior recognition methods have challenges for high recognition accuracy with datasets with problems such as scenes with blurred pictures, and inconsistent objects. To address this problem, we propose a feature pyramid shared convolutional and use the Rep Ghost fusion Cross Stage Partial and Efficient Layer Aggregation Network (RGCSPELAN) to improve the network performance architecture based on improved YOLOv11 for classroom action recognition, called YOLO-RF (as shown in Fig. 2).

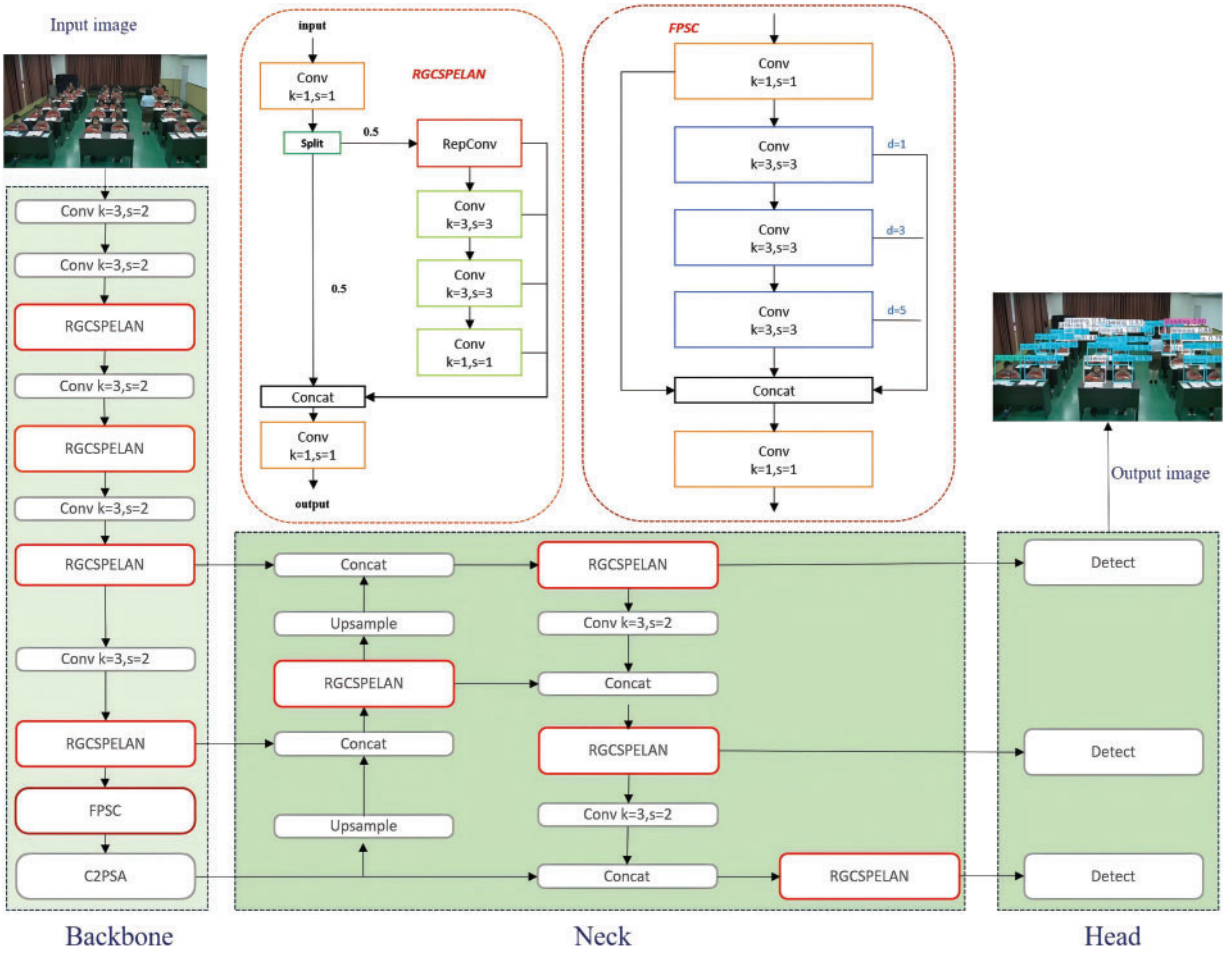


Figure 2: Overview of the RFNet architecture

3.1 Feature Pyramid Shared Convolutional Model

The feature pyramid shared convolutional (FPSC) module is designed for efficient multi-scale feature extraction by leveraging convolutional layers with varying dilation rates. Low dilation rates capture fine-grained local details, while high dilation rates focus on a broader global context, effectively handling features of different scales and contexts in the image. By utilizing a shared convolutional layer, the module significantly reduces the number of trainable parameters compared to using independent convolutional layers for each dilation rate. This parameter-sharing strategy minimizes redundancy and enhances both storage and

computational efficiency. In addition, the module employs 1 × 1 convolutional layers for efficient channel transformations and feature fusion, retaining critical feature information while reducing the parameter count. Unlike pooling-based methods such as SPPF, which may lose detail, the convolution parameter-sharing-based design of FPSC ensures the extraction of finer-grained features. This approach provides greater flexibility and expressive power in capturing intricate details and complex patterns within images.

Fig. 3 shows the proposed FPSC architecture. The d represents dilations, a list of dilation rates for convolutions, and a default value in [1, 3, 5] to control spatial resolution. The FPSC model can be described by the following Eqs. (1)–(5).

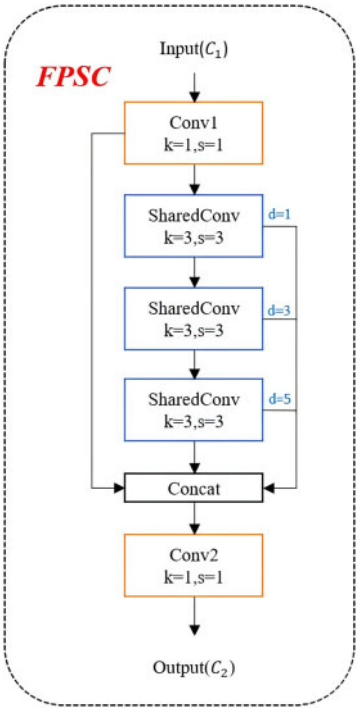


Figure 3: The illustration of FPSC model

First, use the $Conv1$ channel compression for the number of channels to $C_1/2$. It means the input channel count is halved to get the hidden channel size:

$$y_0 = \text{Conv1}_{1\times1}(x, W_1, b_1), y_0 \in \mathbb{R}^{C_1/2\times H\times W} \tag{1}$$

where x represents the input features with a shape of (B, C_1, H, W), W_1 is the weight of the 1×1 convolution, with a shape of $(C_1, C', 1, 1)$, b_1 denotes the bias term. The output y_0 has a shape of (B, C', H, W), where C' is equal to $\frac{C_1}{2}$, B is represents the $bach\_size$, the term H and W represent the feature map's Height and Width. The spatial height of the feature map indicates the number of rows in the 2D representation of the input/output data. The spatial width of the feature map indicates the number of columns in the 2D representation of the input/output data.

Second, shared convolution for each d, the d is dilation rate uses shared convolutional weight, each convolution is applied with different dilation rates to capture multiscale features, as Formula (2), this process generates multiple feature maps iteratively $y_1, y_2, \ldots, y_N$:

$$y_i = \text{SharedConv}_{3\times3}\left(y_{i-1}, W_s, b_s, \text{dilation} = d_i, \text{padding} = \frac{d_i(3-1)}{2}\right) \tag{2}$$

where W_s is the shared 3×3 convolution kernel with shape $(C', C', 3, 3)$. b_s is the bias term, which is set to zero. d_i represents the dilation rate, typically chosen from the set $\{1, 3, 5\}$. The padding is calculated as Eq. (3). The Formula (3) adds 1 to the standard padding calculation, which is typically done to maintain specific tensor dimensions. This adjustment is to compensate for rounding errors in the computation or ensure proper alignment of the output dimensions.:

$$\text{padding} = \frac{d(3-1)+1}{2} \tag{3}$$

Third, concatenation all feature maps along the channel dimension. The concatenation operation is applied along the channel dimension, combining the feature maps obtained from different dilation rates. The resulting feature map is computed as follows (4):

$$Y_{\text{concat}} = \text{Concat}(y_0, y_1, \ldots, y_N) \tag{4}$$

where the concatenated output Y_{concat} has a shape of $(B, C' \times (1+N), H, W)$, with N representing the number of dilation rates applied in the previous step. This process effectively integrates multi-scale features, enriching the representation capacity of the model.

Finally, 1×1 convolution integration, the formula as (5):

$$y_{\text{out}} = \text{Conv2}_{1\times1}(Y_{\text{concat}}, W_2, b_2) \tag{5}$$

where W_2 denotes the weight of the 1×1 convolution kernel with a shape of $(C' \cdot (1+N), 1, 1)$. The b_2 represents the bias term. The final output y_{out} has the shape (B, C_2, H, W), where C_2 is the target number of output channels.

This final convolution operation is used to aggregate the concatenated feature maps and produce the final output with C_2 channels.

For FPSC model, the main idea is to get the first layer of convolution and output a number of channels by inputting the feature map shape as B, C_1, H, W, and then go through the expansion convolution and channel splicing to finally concatenation realize the dimensionality reduction output.

In this paper, the FPSC model is designed to capture multi-scale contextual features efficiently by employing shared convolutional weights across dilated convolutions with varying dilation rates, thereby reducing parameter count and enhancing flexibility for different tasks.

3.2 Rep Ghost Fusion Cross Stage Partial and Efficient Layer Aggregation Model

Referring to the idea in GhostNet [23], mainstream CNN computations contain redundant intermediate feature mappings, which are partially generated through a low-cost operation to reduce computation and parameter count. The commonly used BottleNeck in YOLOv5 and YOLO11 is discarded, and to compensate for the performance loss caused by discarding the residual block, RepConv is used on the gradient circulation branch as a way to enhance the ability of feature extraction and gradient circulation, and RepConv can be fused at the time of inference, which is a two-for-one solution. In this work, design a Rep Ghost fusion Cross Stage Partial and Efficient Layer Aggregation, called RGCSPELAN, the architectural of RGCSPELAN as Fig. 4 shows. The size of RGCSPELAN can be controlled by a scaling factor so that it can accommodate both small and large models.

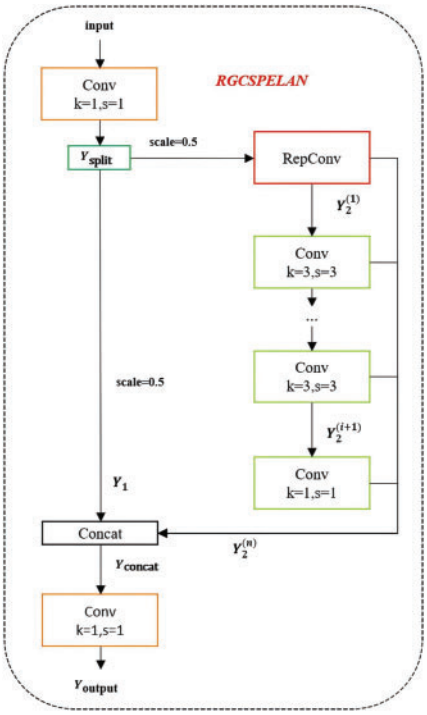


Figure 4: The illustration of RGCSPELAN model

The RGCSPELAN is a convolutional neural network module, it is suitable for extracting multi-scale features and achieving enhancement of feature representation through operations such as concatenation, feature segmentation, and feature transformation. The RGCSPELAN model computation process consists of several steps feature compression, intermediate feature transformation, final convolution, feature splicing, and output computation. The formula is expressed as follows (6)–(12) separately. First, initialize the parameters and sub-modules of the model and then calculate the number of channels in the hidden layer C_{hidden} based on the number of output channels C_2 and the scale factor e.

$$Y_{split} = \text{Conv}_{1\times1}(X, W_1, b_1) \tag{6}$$

where X is the input tensor of shape (B, C_1, H, W), W_1 is the weight of the 1×1 convolution with shape $(C_1, 2C', 1, 1)$, b is the bias term, and Y_{split} has shape $(B, 2C', H, W)$.

Second, feature splitting to calculate the number of intermediate layer channels according to the scaling factor scale, and the scale value default is 0.5.

$$Y_1, Y_2 = \text{Split}(Y_{split}) \tag{7}$$

Third, feature transformation path. The second split Y_2 is processed through a series of convolutional layers, the formula as (8)–(10) show:

$$Y_2^{(1)} = \text{RepConv}(Y_2, W_{rep}, b_{rep}) \tag{8}$$

where Y_2 is the input feature map, $Y_2^{(1)}$ is the output feature map after applying the RepConv operation, RepConv$(\cdot)$ is a convolution operation with re-parameterization, W_{rep} is the convolution weight used in the RepConv layer, and b_{rep} is the bias term used in the $RepConv$ layer.

$$Y_2^{(i+1)} = \text{Conv}_{3\times3}(Y_2^{(i)},\ W_i,\ b_i) \quad \text{for } i = 1, 2, \ldots, n-1 \tag{9}$$

where $Y_2^{(i)}$ represents the input feature map at the $i-th$ stage, $Y_2^{(i+1)}$ output feature map at the $(i+1)-th$ stage. $\text{Conv}_{3\times3}$ represents standard 3×3 convolution operation. W_i is the weight of the convolution filter at stage i, b_i is the bias term for the convolution at stage i, and i is the index of the convolutional layer, iterating through the network stages.

$$Y_2^{(n)} = \text{Conv}_{1\times1}(Y_2^{(n-1)}, W_{final}, b_{final}) \tag{10}$$

where $Y_2^{(n-1)}$ represents the input feature map from the previous convolutional layer, and $Y_2^{(n)}$ represents the final output feature map after the last convolutional operation. $\text{Conv}_{1\times1}(\cdot)$ is a 1×1 convolution operation used to combine features and reduce the number of channels. W_{final} is the weight of the final convolutional layer, and b_{final} is the bias term for the final convolutional layer.

These formulas collectively represent the hierarchical feature transformation process within the RGCSPELAN module, progressively refining feature maps through convolutional layers with different kernel sizes and operations. The intermediate feature maps are progressively transformed through stacked convolutional layers.

Then, concatenated the feature map. The concatenated feature map has the shape $(B, C' + mid \cdot (n+1), H, W)$, as Formula (11) describe:

$$Y_{\text{concat}} = \text{Concat}(Y_1,\ Y_2^{(n)}) \tag{11}$$

Finally, there is a convolutional output Y_{output}, as Formula (12). This step integrates the concatenated feature maps and produces the final output of shape $(B,\ C_2,\ H,\ W)$, after after applying a 1×1 convolution operation:

$$Y_{output} = \text{Conv}_{1\times1}(Y_{concat}, W_2, b_2) \tag{12}$$

where B represents the batch size, C_2 denotes the number of output channels, H and W are the spatial dimensions (height and width). $\text{Conv}_{1\times1}(\cdot)$ is a 1×1 convolution operation, used to aggregate feature information across channels and reduce the number of feature channels efficiently. Y_{concat} is the concatenated feature map obtained from multiple dilation convolutions, with dimensions $(B, C' \cdot (1+N), H, W)$, where C' is the intermediate number of channels and N is the number of dilation rates applied. The W_2 is the weight matrix of the 1×1 convolutional layer, with dimensions $(C' \cdot (1+N), C_2, 1, 1)$. b_2 represents the bias term added during the convolution process to adjust the output features, with dimensions $(C_2,)$.

The RGCSPELAN is a multilayer convolutional network with dynamic channel segmentation, feature extraction and splicing. The feature flow is enhanced by utilizing segmentation and splicing operations, while the expressive power of the network is improved by intermediate convolutional blocks.

The RGCSPELAN module is designed to efficiently process input features by splitting, applying convolutional operations, and concatenating intermediate feature representations to generate refined outputs. The structure consists of an initial 1×1 convolution for channel reduction, a split operation to divide the feature map, a sequence of convolutional layers, and a final 1×1 convolution to produce the desired output.

4 Experiments

4.1 Datasets

In this work, we used the STBD-08 [5] dataset and SCB dataset3 [24] to evaluate the effectiveness and robustness of our proposed RFNet method.

4.2 STBD-08 Dataset

The Student Teacher Behavior Dataset (STBD-08) [5] includes eight typical classroom behaviors: writing, reading, listening, turning around, raising hands, standing, discussing, and guiding. It captures teacher and student actions in a classroom setting. A total of 131 class videos from China's national public resource platform (2019 primary school) were collected and processed using FFmpeg for automatic segmentation. After sorting and labeling, the dataset was compiled into 4432 images with 151,574 bounding boxes [5]. The STBD-08 dataset is created by Zhao and Zhu [5], it is publicly available from the github link provided by the author. In this work, we divide it with a ratio of 4:1 as training and validation sets to evaluate the performance of the proposed YOLO-RF method.

4.3 SCB-Dataset3

The Student Classroom Behavior dataset (SCB dataset3) [24] is a public dataset, which contains 18.4 thousand annotations across 4.2 thousand images, encompassing three behavioral categories: hand-raising, reading, and writing. To evaluate the performance of the proposed framework, the dataset was partitioned into training and validation sets in a 4:1 ratio.

4.4 Experiment Setting

We used the STDB-08 dataset and SCB dataset 3 in our experiments. Our proposed YOLO-RF model was trained on an RTX 4090 GPU with 24 GB memory on Ubuntu 20.4 operating system, with a batch size of 32 and 4 workers. We employed SGD as the optimizer with a learning rate of 0.01, a close mosaic of 0.0, the input images size are 640 × 640, a momentum of 0.937, a warmup momentum of 0.8, a warmup bias learning rate of 0.1, and a weight decay of 0.0005. The model was trained for 300 epochs without pre-training.

4.5 Evaluation Indicators

This study evaluates the model's performance across four indicators: parameter count (Params), computational complexity (GFLOPs), FPS (Frames Per Second), mean average precision at 0.5 (mAP@0.5) and at 0.5–0.9 (mAP@0.5–0.9). Params quantifies the model's size, while GFLOPs measure its computational complexity. Lower Params and GFLOPs indicate reduced computational demands, enabling deployment on low-end devices with modest hardware capabilities. FPS represents the number of frames processed per second, with higher values signifying superior real-time performance. mAP@0.5 assesses the model's detection accuracy across all categories in the test set, with higher scores reflecting better overall detection performance. The formula for precision (P) as the Eq. (13) shows, is used in mAP@0.5 and mAP@0.5–0.9 computation, and is defined as the ratio of true positive samples (TP) to the sum of true positive and false positive samples (TP+FP).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

Recall (R) is defined as the proportion of correctly identified defect samples (TP) to the total number of actual defect samples (TP+FN). The equation is described as follows (14):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

where AP@0.5 represents the average precision of a certain type of defect when the IOU threshold is set to 0.5. mAP@0.5 represents the mean of the average precision across all types of defects. mAP@0.5–0.9 refers to the mean average precision calculated across multiple IoU (Intersection over Union) thresholds, ranging from 0.5 to 0.9, typically in increments of 0.05. This metric provides a comprehensive evaluation of the model's detection accuracy by averaging performance over varying levels of overlap criteria between predicted and ground truth bounding boxes. What's more, higher mAP@0.5–0.9 values indicate better overall detection consistency across different IoU thresholds. The AP and mAP@0.5 as the following Eqs. (15) and (16):

$$\text{AP@0.5} = \frac{1}{n}\sum_{i=1}^{n} \text{P}^i \tag{15}$$

$$m\text{AP@0.5} = \frac{1}{n}\sum_{k=1}^{n} \text{AP@0.5}k \tag{16}$$

4.6 Ablation Experiments

An ablation experiment was conducted to assess better the impact of improved modules and their combination on enhancing the original model's performance. The experimental results are presented in Table 1, evaluating the contributions of different components to the model's performance on the STDB-08 dataset. The inclusion of feature pyramid shared convolutional (FPSC) and Rep Ghost fusion Cross Stage Partial and Efficient Layer Aggregation (RGC-SPELAN) is systematically analyzed across four configurations. The baseline configuration achieves a mean Average Precision (mAP) of 91.8% at 0.5 IoU (mAP@0.5), with a model size of 5.2 MB, 2.6 million parameters, 6.3 Giga Floating-point Operations Per Second (GFLOPs), and an inference speed of 518.3 Frames Per Second (FPS). Adding RGC-SPELAN to the baseline improves the mAP slightly to 91.9% but increases the size (5.5 MB) and parameters (Params) (2.7 M) while reducing the FPS to 365.1. Incorporating FPSC instead leads to a slightly reduced mAP of 91.7% but improves computational efficiency (size: 4.6 MB, FPS: 601.5). The full model, combining both FPSC and RGC-SPELAN, achieves the best overall balance with the highest mAP of 92.0%, reduced size (4.9 MB), fewer parameters (2.4 M), and high computational efficiency (6.2 GFLOPs, 608.8 FPS). This demonstrates the synergistic effect of FPSC and RGC-SPELAN, leading to improved accuracy and computational efficiency. What's more, the he full model, combining both FPSC and RGCSPELAN, 328 achieves the best preceision value (P) 87.2% with slightly the time delay (Latency) 1.64 ms, the Recall 329 value (R) is 86.2% and the network layer number (Layers) only 192, less than the baseline 238.

The normalized confusion matrix of the proposed model on the STBD-08 dataset as Fig. 5 shows illustrates the performance and highlights its ability to classify various activities with high accuracy. Diagonal values, representing correctly classified instances, are consistently high, with notable precision for activities such as "writing" (0.95), "listening" (0.95), and "discussing" (0.93). However, some misclassifications are observed, such as "listening" being confused with "background" (0.18) and "reading" (0.07). These off-diagonal values suggest areas for improvement, particularly in distinguishing similar activities or handling background noise. The model demonstrates robust performance while indicating potential refinements for challenging activity classifications.

Table 1: Results of the ablation experiments on the STBD-08 dataset

Method	FPSC	RGCSPELAN	P (%)	R (%)	mAP @0.5 (%)	Size (M)	Params (M)	GFLOPs	FPS	Latency (ms)	Layers
YOLOv11 (Baseline)			86.4	87.2	91.8	5.2	2.6	6.3	518.3	1.93 ± 0.02	238
+FPSC	√		86.7	**87.3**	91.9	5.5	2.7	6.3	365.1	2.74 ± 0.70	238
+RGCS PELAN		√	86.9	86.3	91.7	**4.6**	**2.2**	**6.2**	601.5	1.66 ± 0.02	**192**
+FPSC+ RGCS PELAN	√	√	**87.2**	86.2	**93.1**	4.9	2.4	**6.2**	**608.8**	**1.64 ± 0.02**	**192**

Note: Bold values indicate the best performance for each metric across all methods.

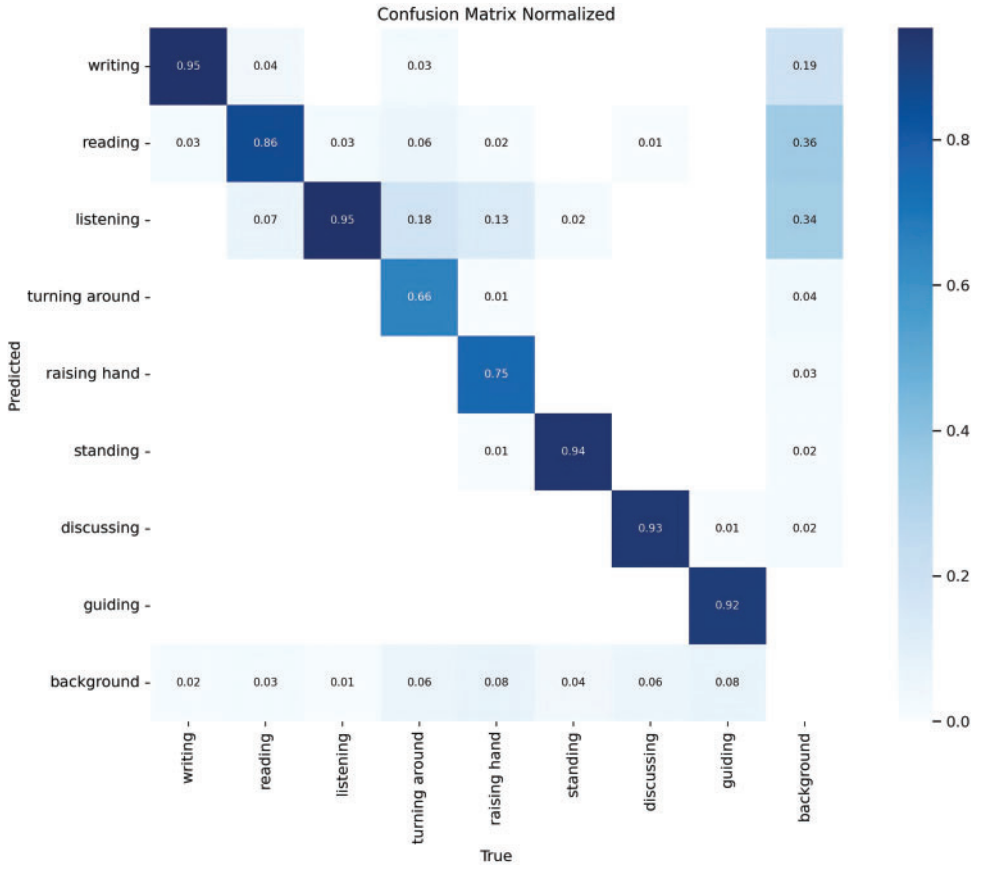


Figure 5: Confusion matrix of the proposed model on STBD-08 dataset

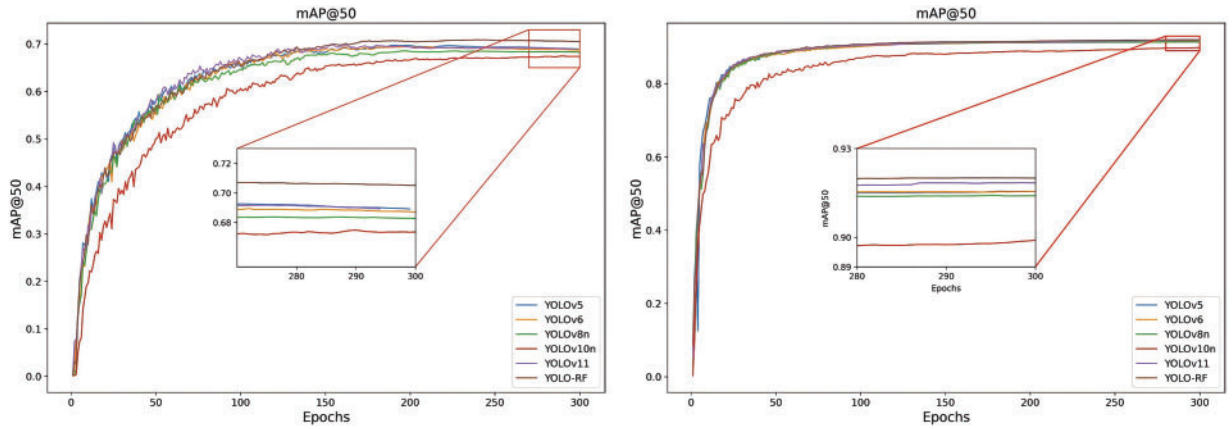
4.7 Comparision Experiment and Analysis

Table 2 is a concise architectural comparison between SPPF, ASPP, and the proposed FPSC module. While SPPF and ASPP utilize fixed pooling and dilated convolutions respectively with simple concatenation, FPSC introduces a shared convolutional strategy combined with fusion, achieving comparable computational cost (6.2 GFLOPs) and effectively replacing SPPF in the network with enhanced feature aggregation.

Table 2: Concise architectural comparison between SPPF, ASPP and FPSC

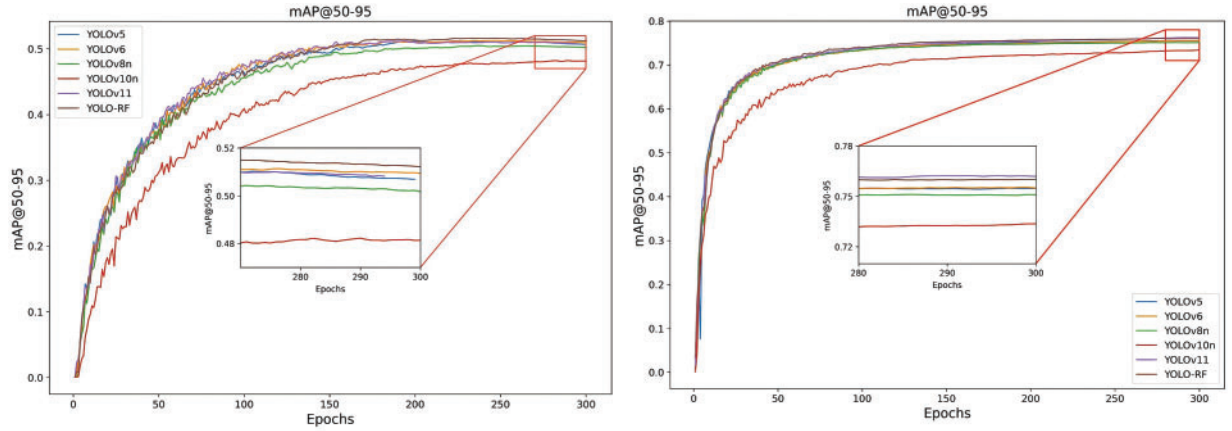
Model	Receptive field strategy	Feature fusion	GFLOPs	Position in network
SPPF	Fixed pooling pyramid	Concatenation	6.3	YOLOv5 Neck
ASPP	Dilated convolutions	Concatenation	10.2	Head
FPSC (Ours)	Shared convolutional	Shared conv + fusion	6.2	Replacing SPPF

Figs. 6 and 7 compare mAP@50 and mAP@50–95 performance across various detection models over 300 training epochs on the SCB dataset3 and the STDB-08 dataset, respectively. The proposed method (labeled "YOLO-RF") demonstrates superior convergence behavior, achieving the highest mAP@50 and mAP@50–95 value compared to YOLOv5, YOLOv6, YOLOv8, YOLOv10n, and YOLOv11. Notably, the proposed approach converges faster than YOLOv10n, which exhibits slower progress and lower ultimate accuracy. Models like YOLOv6 and YOLOv8 perform comparably in early and mid-training stages but are ultimately outperformed by the proposed method. This figure highlights the efficiency and robustness of the proposed model in achieving higher detection accuracy while maintaining competitive training dynamics. Fig. 8 presents the precision-recall (PR) curves for individual activity classes and the overall performance of the proposed method. The model achieves an overall mean Average Precision at 0.5 (mAP@0.5) of 0.920, with strong class-specific precision values. Notably, activities such as "writing" (0.983), "listening" (0.982), and "standing" (0.958) exhibit high PR curve performance, reflecting their reliable detection. However, relatively lower performance is observed for "raising hand" (0.809) and "turning around" (0.803), indicating these classes pose greater challenges, likely due to increased variability in their representation. The blue curve, representing all classes, demonstrates the robustness of the YOLO-RF model across diverse activities, ensuring consistent precision and recall trade-offs. Fig. 9 presents the Precision-Recall (PR) curves for the proposed method on SCB dataset3, showing performance across three specific behavioral categories: hand-raising, reading, and writing, as well as an aggregated curve for all classes. The individual PR curves are color-coded, with the hand-raising class achieving the highest precision-recall performance, reflected by a mean Average Precision (mAP) of 0.800. This is followed by reading, which attained an mAP of 0.733 while writing demonstrated relatively lower performance with an mAP of 0.591. The PR curve for all classes, shown as the bold blue line, indicates the overall detection performance across the dataset, achieving a mean Average Precision (mAP@0.5) of 0.708. The results demonstrate that the proposed YOLO-RF method performs effectively, particularly for behaviors with distinct visual features, while still providing robust detection across all behavior categories. Table 3 reports the average precision (AP) of the model across different Intersections over Union (IoU) thresholds, ranging from 50% to 95%. The AP at a threshold of 50% is the highest (91.44%), indicating strong performance at lower IoU requirements. However, as the IoU threshold increases, the AP gradually decreases, reflecting the model's diminishing ability to detect bounding boxes under stricter localization criteria accurately. Specifically, the AP drops to 89.84% at 70%, 82.13% at 80%, and decreases to 44.71% at 90%. At the most stringent threshold of 95%, the AP is significantly reduced to 7.38%. These results demonstrate the model's robustness at moderate IoU thresholds while revealing challenges in achieving exact localization under stricter conditions.



(a) mAP@50 comparison results on the SCB dataset3. (b) mAP@50 comparison results on the STBD-08 dataset.

Figure 6: Comparison of mAP@50 with previous methods. **(a)** Experiment results on the SCB dataset3. **(b)** Experiment results on the STBD-08 dataset



(a) mAP@50-95 comparison results on the SCB dataset3. (b) mAP@50-95 comparison results on the STBD-08 dataset.

Figure 7: Comparison of mAP@50–95 with previous methods. **(a)** Experiment results on the SCB dataset3. **(b)** Experiment results on the STBD-08 dataset

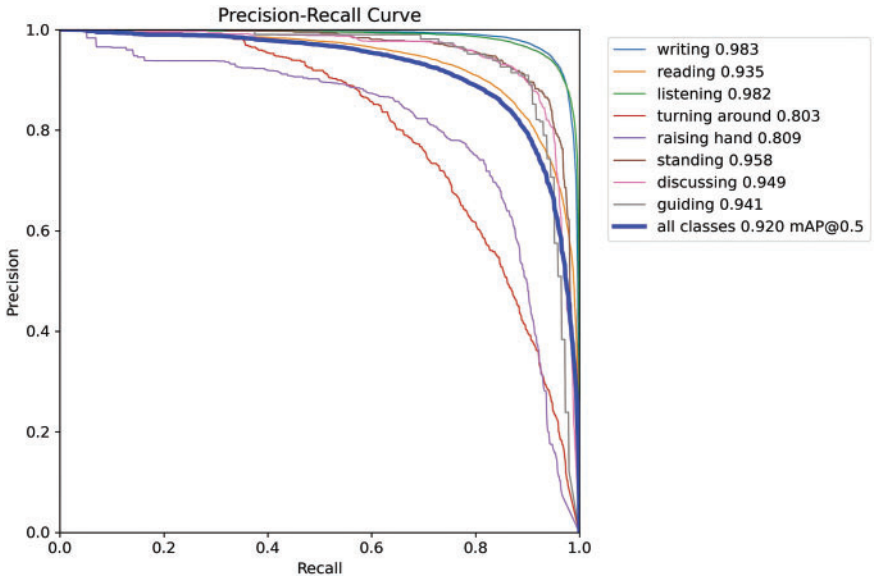


Figure 8: The precision recall curve of the proposed YOLO-RF method on STBD-08 dataset

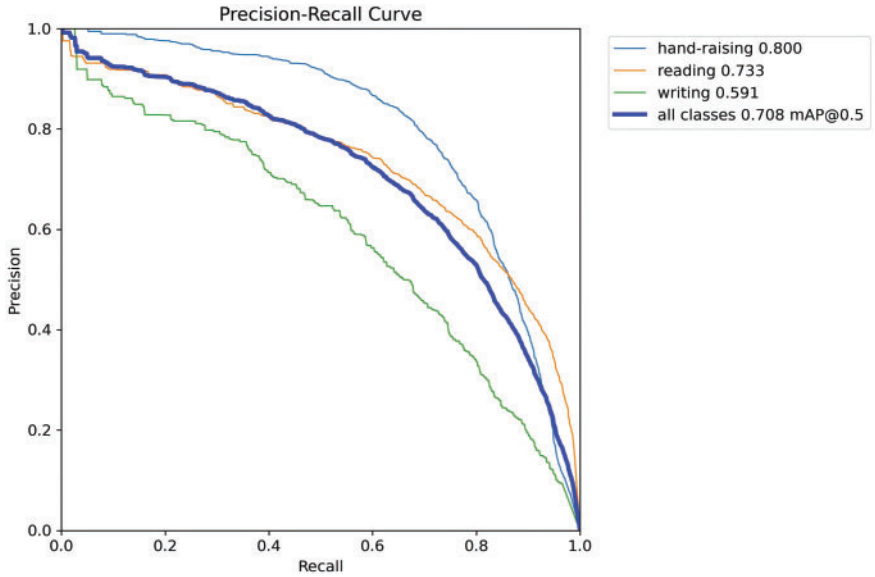


Figure 9: The precision recall curve of the proposed YOLO-RF method on SCB dataset3

Table 3: Experiment results of bbox AP@[50–95] values on the STBD-08 dataset

Thresh	50	55	60	70	75	80	85	90	95
AP	91.44	91.36	91.21	89.84	87.53	82.13	69.41	44.71	7.38

Table 4 compares the performance of various YOLO versions and the proposed method on the STBD-08 dataset in terms of precision, recall, F1-score, GFLOPs, mean Average Precision (mAP), model size, network depth, parameter count, latency, speed, and frame rate (fps).

Table 4: Comparison of methods on the STBD-08 dataset with various evaluation indicators

Methods	P	R	F1	GFLOPs	mAP @50 (%)	Size (M)	Layers	Params (M)	Latency (ms)	Speed (ms)	FPS
Faster R-CNN [3]	80.3	82.1	80.6	28.5	79.9	152.3	231	10.8	4.1 ± 1.3	7.3	68.8
YOLOv5 [4]	87.5	86.1	**87.0**	7.1	91.5	5.1	193	2.5	1.58 ± 0.06	1.7	634.5
YOLOv6 [20]	88.0	86.2	**87.0**	11.8	91.6	8.3	**142**	4.2	**1.27 ± 0.05**	2.2	**785.9**
YOLOv8n [21]	87.0	86.1	**87.0**	8.1	91.4	6.0	168	3.0	1.41 ± 0.08	1.6	482.1
YOLOv10n [22]	84.5	84.8	84.0	8.5	89.9	8.5	229	2.3	1.97 ± 0.09	**1.3**	476.7
YOLOv11 [9]	86.4	**87.2**	**87.0**	6.3	91.8	5.2	238	2.4	1.93 ± 0.02	1.9	518.3
RTDETR-R18 [25]	88.6	86.6	86.3	57.0	90.7	307.5	299	**2.0**	1.48 ± 1.01	5.4	67.4
RTDETR-R50 [25]	**89.0**	86.7	83.9	129.6	91.4	654.6	479	4.2	3.0 ± 1.6	8.6	33.8
RFNet	87.2	86.2	**87.0**	**6.2**	**93.1**	**4.9**	192	2.4	1.64 ± 0.02	2.2	608.8

Note: Bold values indicate the best performance for each metric across all methods.

Among all evaluated methods, the proposed RFNet achieves the highest mAP@50 of 93.1%, preceision 87.2%, recall value 86.2%, F1 confidence value (F1) 87.0, while maintaining a relatively small model size of 4.9 M and a fast inference speed of 608.8 FPS, demonstrating an excellent trade-off between accuracy and efficiency. In contrast, larger models like RTDETR-R50 and RTDETR-R18 achieve lower mAP@50 (91.4% and 90.7%, respectively), with significantly higher Giga Floating-point Operations Per Second (GFLOPs) (129.6 and 57.0) and slower inference speeds (33.8 and 67.4 FPS), indicating their limitations for real-time classroom applications. The YOLOv6, YOLOv8n, and YOLOv10 variants exhibit competitive performance in both accuracy and speed, but RFNet outperforms them with a higher mAP and lower latency. Notably, although YOLOv6 achieves the highest FPS (785.9), its mAP@50 is lower than that of RFNet. Faster R-CNN, as a two-stage detector, exhibits the lowest inference speed (68.8 FPS) and the highest latency 4.1 ms and 393 the speed 7.3 ms, while also achieving the lowest mAP@50 (79.9%), confirming its unsuitability for real-time deployment in this context.

In summary, the comparison highlights YOLO-RF as the most accurate model. YOLO-RF maintains a strong balance between accuracy (highest mAP@50), model size, and computational efficiency, making it a robust and lightweight alternative.

The comparative analysis in Table 5 underscores the efficacy of the proposed method RFNet ("YOLO-RF") on the SCB-dataset3 across key performance metrics. Among the examined methods, our approach achieves the highest mAP@50 of 71.0%, surpassing the popular YOLOv5 (69.7%), YOLOv6 (69.1%), YOLOv8n (68.5%), YOLOv10n (67.5%), and YOLOv11 (69.6%). The YOLO-RF approach has the highest precision at 68.6%, surpassing the baseline method (YOLOv11) at 3.3% the F1 score also achieves the highest value at 71.0%. Notably, this improvement is attained with minimal computational and model complexity.

In terms of computational cost, our method demonstrates a significant advantage by utilizing only 6.2 GFLOPs, the lowest among all compared methods, including YOLOv6 (11.8 GFLOPs) and YOLOv8n (8.1 GFLOPs), which require substantially higher computational resources. Furthermore, our model is lightweight, with a size of 4.9 MB, outperforming its counterparts such as YOLOv6 (8.3 MB) and YOLOv8n

(6.0 MB). This reduction in model size is achieved while maintaining competitive architectural depth (192 layers) and a lower parameter count (2.4 M), compared to YOLOv11 (2.6 M) and YOLOv6 (4.2 M).

Table 5: Performance of compared methods on SCB-dataset3 with various evaluation indicators

Methods	P	R	F1	GFLOPs	mAP @50 (%)	Size (M)	Layers	Params (M)	Latency (ms)	Speed (ms)	FPS
Faster R-CNN [3]	67.3	67.1	67.0	28.5	59.3	152.3	231	13.8	4.6 ± 2.2	15.0	69.4
YOLOv5 [4]	63.9	67.5	66.0	7.1	69.7	5.0	193	2.5	1.91 ± 0.64	1.7	524.5
YOLOv6 [20]	65.0	65.5	66.0	11.8	69.1	8.3	**142**	4.2	1.78 ± 0.80	1.8	562.0
YOLOv8n [21]	65.3	65.0	65.0	8.1	68.5	6.0	168	3.0	1.47 ± 0.30	1.8	**678.8**
YOLOv10n [22]	63.3	64.5	64.0	6.5	67.5	5.5	229	**2.3**	2.43 ± 0.64	**1.2**	411.3
YOLOv11 [9]	65.3	66.9	64.0	6.3	69.6	5.2	238	2.6	1.93 ± 0.02	1.8	519.2
RTDETR-R18 [25]	**73.0**	68.7	**68.0**	57.0	**71.7**	307.4	229	19.9	**0.7 ± 0.08**	3.1	139.3
RTDETR-R50 [25]	70.6	**70.4**	67.3	70.5	71.4	654.5	479	50.0	2.3 ± 1.3	9.1	42.6
RFNet	68.6	65.9	67.0	**6.2**	71.0	**4.9**	192	2.4	2.45 ± 0.76	1.8	408.5

Note: Bold values indicate the best performance for each metric across all methods.

Among all methods, RTDETR-R18 achieves the highest mAP@50 of 71.7%, but at the cost of significantly larger model size (307.4 M), higher GFLOPs (57.0), and a much lower FPS (139.3), making it less suitable for real-time or resource-constrained scenarios. RTDETR-R50 similarly delivers strong F1 performance (70.5) but exhibits the lowest FPS (42.6) and highest latency (2.3 ms), indicating limited efficiency. In contrast, the proposed RFNet achieves a competitive mAP@50 of 71.0%, while offering a lightweight model size (4.9 M) and a high inference speed of 408.5 FPS. It also maintains a balanced F1-score of 67.0, the highest among all YOLO-based detectors, indicating strong detection capability without compromising speed or computational cost. YOLOv8n records the highest FPS (678.8), but its mAP@50 (68.5%) and F1-score (65.0) are noticeably lower than RFNet. Faster R-CNN shows poor real-time performance, with high latency (4.6 ms), low FPS (69.4), and the lowest mAP@50 (59.3%) among all methods.

Overall, RFNet demonstrates an excellent trade-off between detection accuracy, speed, and model compactness, confirming its effectiveness and practicality for real-time classroom behavior recognition tasks in resource-constrained environments.

4.8 Visible Results

Visible results on the STBD-08 dataset, the comparative detection performance of YOLOv10n, YOLOv11, and YOLO-FR, demonstrates the progressive improvement in recognizing and classifying activities such as listening, writing, and standing across different scenarios, as Fig. 10 shows.

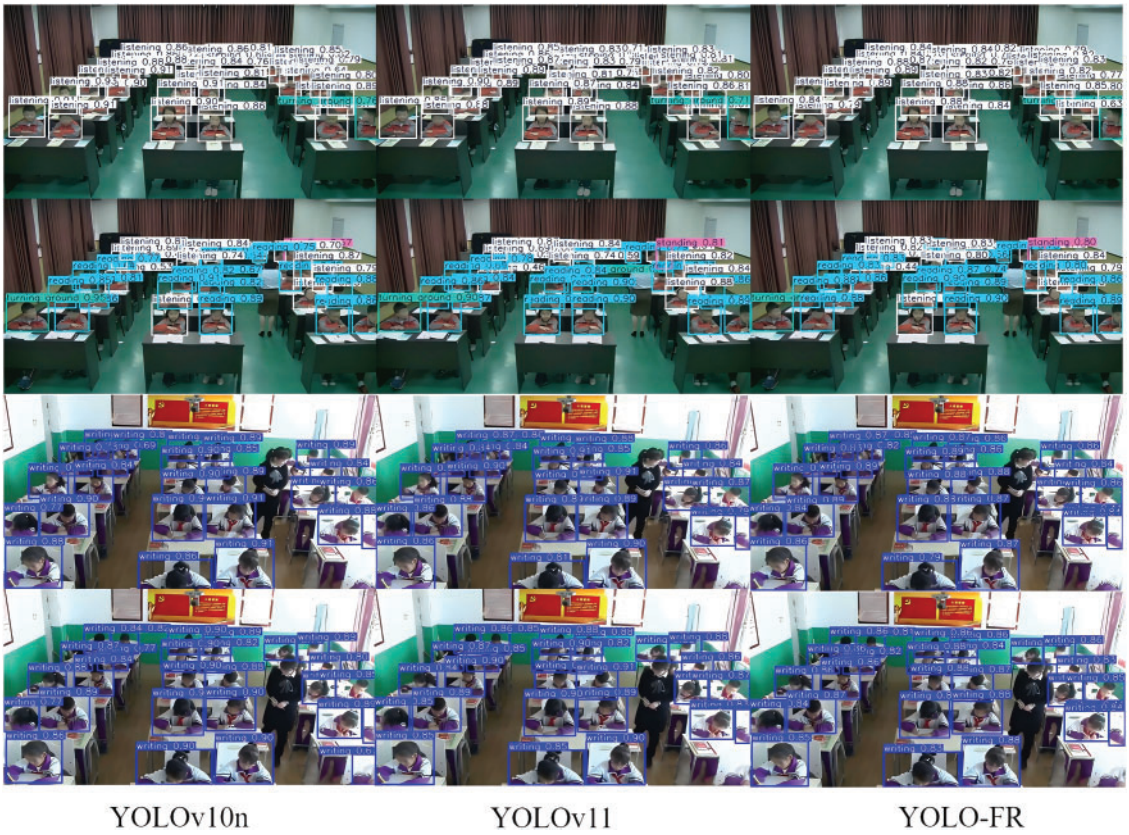


Figure 10: Visual comparison of detection results between YOLOv10n, YOLOv11, and our proposed methods

5 Conclusion

In this work, we propose an effective lightweight and a robust alternative object detector method YOLO-FR, called RFNet, to address the challenge of student classroom action recognition in scenes with blurred pictures, and inconsistent objects. We got benchmark tests on YOLOv5, YOLOv6, YOLOv8n, YOLOv10n and YOLOv11. RFNet demonstrates a slight improvement in mAP@50 and inference speed, along with reduced latency and architectural complexity compared to YOLOv11, making it a more efficient alternative while maintaining comparable detection performance. Our proposed improved YOLOv11 algorithm provides a novel improvement and a lightweight with a lower parameter count (2.4 M) for future classroom action recognition studies. However, this study evaluates performance on only two datasets, both relatively small and domain-specific, potentially limiting the generalizability of the results. In future work, we plan to explore more diverse and larger-scale datasets, including cross-domain and multi-institution classroom scenarios, to comprehensively evaluate the generalization ability of the model.

Furthermore, owing to the overlap observed in certain behaviors, we plan to incorporate pose-estimation methods and multi-scale feature maps by randomly masking contiguous regions [26] to delineate more nuanced categories, including activities such as reading, writing, and listening. We anticipate that this study will foster more comprehensive research into classroom attentiveness and offer enhanced tools and methodologies to elevate the quality of teaching and learning.

To further enhance deployment flexibility, future work will focus on optimizing the model for edge environments. We plan to apply model compression techniques such as quantization, pruning, and knowledge

distillation to reduce computational costs. These methods will allow the model to be deployed on lower-end devices like Jetson Nano while maintaining acceptable accuracy. We also aim to expand evaluation to additional datasets from different institutions to further verify generalizability across diverse classroom environments and conduct a full ablation study against ASPP, SPPF and so on.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Fundamental Research Grant Scheme (FRGS) of Universiti Sains Malaysia, Research Number: FRGS/1/2024/ICT02/USM/02/1.

Author Contributions: The authors confirm their contribution to the paper as follows: wrote the first draft of the manuscript and preparation: Chuanchuan Wang; analyzed and developed the main idea of the proposed framework: Xiao Yang; results analysis: Hao Zhang; data collection: Xiang Li; review, and editing: Ahmad Sufril Azlan Mohamed and Mohd Halim Bin Mohd Noor. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The experimental datasets essential for replicating the methodological outcomes are presently restricted from public access due to their continued utilization in concurrent scholarly investigations.

Ethics Approval: None.

Conflicts of Interest: The contributors affirm the absence of competing financial interests or personal relationships that might influence the outcomes of this investigation.

References

1. Zhou L, Liu X, Guan X, Cheng Y. CSSA-YOLO: cross-scale spatiotemporal attention network for fine-grained behavior recognition in classroom environments. Sensors. 2025;25(10):3132. doi:10.3390/s25103132.
2. Dang M, Liu G, Li H, Xu Q, Wang X, Pan R. Multi-object behaviour recognition based on object detection cascaded image classification in classroom scenes. Appl Intell. 2024;54(6):4935–51. doi:10.1007/s10489-024-05409-x.
3. Gao C, Ye S, Tian H, Yan Y. Multi-scale single-stage pose detection with adaptive sample training in the classroom scene. Knowl-Based Syst. 2021;222:107008. doi:10.1016/j.knosys.2021.107008.
4. Tang L, Xie T, Yang Y, Wang H. Classroom behavior detection based on improved YOLOv5 algorithm combining multi-scale feature fusion and attention mechanism. Appl Sci. 2022;12(13):6790. doi:10.3390/app12136790.
5. Zhao J, Zhu H. CBPH-net: a small object detector for behavior recognition in classroom scenarios. IEEE Trans Instrum Meas. 2023;72:1–12. doi:10.1109/tim.2023.3296124.
6. Li Y, Qi X, Saudagar AKJ, Badshah AM, Muhammad K, Liu S. Student behavior recognition for interaction detection in the classroom environment. Image Vis Comput. 2023;136:104726. doi:10.1016/j.imavis.2023.104726.
7. Sun JJ, Karigo T, Chakraborty D, Mohanty SP, Wild B, Sun Q, et al. The multi-agent behavior dataset: mouse dyadic social interactions. Adv Neural Info Proc Sys. 2021;2021(DB1):1–15.
8. Dang M, Liu G, Xu Q, Li K, Wang D, He L. Multi-object behavior recognition based on object detection for dense crowds. Expert Syst Appl. 2024;248:123397. doi:10.1016/j.eswa.2024.123397.
9. Jocher G, Qiu J, Chaurasia A. Ultralytics YOLO (Version 8.0.0) [Computer software]. 2023 [cited 2025 Jun 18]. Available from: https://github.com/ultralytics/ultralytics.
10. Gu C, Li Y. Analysis of art classroom teaching behavior based on intelligent image recognition. Mobile Inform Syst. 2022;2022(1):5736407. doi:10.1155/2022/5736407.
11. Wu D, Chen J, Deng W, Wei Y, Luo H, Wei Y. The recognition of teacher behavior based on multimodal information fusion. Math Probl Eng. 2020;2020(1):8269683. doi:10.1155/2020/8269683.
12. Wu Q, Wu Y, Zhang Y, Zhang L. A local-global estimator based on large kernel CNN and transformer for human pose estimation and running pose measurement. IEEE Trans Instrum Meas. 2022;71:1–12. doi:10.1109/tim.2022.3200438.

13. Redmon J. You only look once: unified, real-time object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition; 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA. 2016. p. 779–88.
14. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Ssd: single shot multibox detector. In: Computer Vision-ECCV 2016: 14th European Conference. Amsterdam, The Netherlands; 2016. p. 21–37.
15. Hur P and Bosch N. Tracking individuals in classroom videos via post-processing OpenPose data. In: LAK22: 12th International Learning Analytics and Knowledge Conference. New York, NY, USA. 2022. p. 465–71.
16. Tang L, Gao C, Chen X, Zhao Y. Pose detection in complex classroom environment based on improved Faster R-CNN. IET Image Process. 2019;13(3):451–7. doi:10.1049/iet-ipr.2018.5905.
17. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2016;39(6):1137–49. doi:10.1109/tpami.2016.2577031.
18. Wang Z, Yao J, Zeng C, Wu W, Xu H, Yang Y. YOLOv5 enhanced learning behavior recognition and analysis in smart classroom with multiple students. In: IEEE International Conference on Intelligent Education and Intelligent Research (IEIR). Wuhan, China; 2022. p. 23–9.
19. Zhu H, Zhao J, Niu L. An efficient model for student behavior recognition in classroom. In: IEEE International Conference on Intelligent Education and Intelligent Research (IEIR). Wuhan, China; 2022. p. 142–7.
20. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications. arXiv: 2209.02976,2022.
21. Jocher G. YOLO by Ultralytics. 2023 [cited 2025 Jun 18]. Available from: https://github.com/ultralytics/ultralytics.
22. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J et al. YOLOv10: real-time end-to-end object detection. arXiv:2405.14458, 2024.
23. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA; 2020. p. 1577–86.
24. Yang F, Wang T, Wang X. Student classroom behavior detection based on YOLOv7+ BRA and multi-model fusion. In: International Conference on Image and Graphics (ICIG). Cham: Springer; 2023. p. 41–52.
25. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. Detrs beat yolos on real-time object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA; 2024. p. 16965–74.
26. Yang X, Mohamed ASA, Wang C. ShadowFPN-YOLO: a real-time NMS-free detector for remote sensing ship detection. IEEE Access. 2025;13:55801–14. doi:10.1109/ACCESS.2025.3554018.