https://doi.org/10.65091/icicset.v2i1.29

Real-Time Student Attentiveness Monitoring System using YOLOv5

Rudra Nepal
Nepal College of Information Technology,
Pokhara University, Nepal
rudra.nepal@ncit.edu.np

Rasad Regmi
Nepal College of Information Technology,
Pokhara University, Nepal
regmirasad53@gmail.com

Birat Aryal
Nepal College of Information Technology,
Pokhara University, Nepal
mailbirataryal@gmail.com

Dipesh D.C.
Nepal College of Information Technology,
Pokhara University, Nepal
dipeshdc890@gmail.com

Pranav Subedi
Nepal College of Information Technology,
Pokhara University, Nepal
pranavsubedi12@gmail.com

***Abstract*—This paper presents real-time classroom monitoring system designed to detect and quantify student attentiveness using deep learning and computer vision. By leveraging YOLOv5 for behavior recognition (reading, writing, hand-raising, and focused gaze) and integrating head pose estimation, the system provided objective, dynamic engagement metrics. The model was trained using the SCB-Dataset merged with a custom focus dataset, achieving a precision of 1.00 and mAP@0.5 of 0.681 on validation data. With real-time video stream processing, mobile webcam integration, and a responsive dashboard, ClassCam supported educators in fostering engagement through data-driven decisions.**

***Keywords*—Student Attentiveness, YOLOv5, Computer Vision, Real-Time Monitoring, Education Technology, Behavior Detection**

I. Introduction

Real-Time Student Attentiveness Monitoring System Using YOLOv5 is real-time classroom monitoring system designed to analyze student attentiveness using computer vision. By leveraging YOLOv5, the system detects attentive behaviors such as reading, writing, focusing, and hand raising, and displays them with annotated green bounding boxes in a live video feed. It calculates the attentiveness percentage every minute, based on the number of attentive students detected and a manually entered total student count, providing educators with valuable insights into classroom engagement.

A. Background

In modern education, student attention is one of the major determinants of learning outcomes. Conventional approaches to tracking classroom attention, such as manual observation or self-report, are subjective, unreliable, and time-consuming. With the development of computer vision and artificial intelligence, students' behavior can be automatically tracked from video recordings. Object detection (e.g., YOLOv5) and pose estimation technology allow real-time tracking of students' attentiveness cues like head pose, gaze, hand-raising, reading, and writing. These technologies can be used to facilitate more efficient classroom management and provide teachers with objective and data-driven feedback.

B. Motivation

The motivation behind ClassCam stems from the growing need to enhance student engagement in classrooms through objective, real-time monitoring. Teachers often struggle to assess attentiveness accurately, especially in larger or hybrid learning environments. By using computer vision to automatically detect attentive behaviors, we aim to support educators with actionable insights, reduce manual observation burden, and ultimately improve the quality of learning by identifying when and where students lose focus.

C. Problem Statement

In conventional classroom environments, monitoring student attentiveness is largely dependent on the teacher's personal observation and intuition. This manual process becomes increasingly difficult and ineffective as class sizes grow, leading to potential oversight of disengaged or struggling students. Moreover, teachers often lack real-time tools or data-driven methods to assess how focused students are during lectures.

With the rise of digital tools and remote learning environments, the challenge of accurately gauging student engagement has become even more complex. While there are various tools to track attendance or participation, very few solutions provide insights into actual attentiveness during class sessions. Existing systems either require manual inputs or fail to capture subtle but important behavioral cues such as reading, writing, or hand raising.

There is a clear need for an automated, non-intrusive, and real-time solution that can monitor classroom behavior, identify attentive actions, and quantify attentiveness without disrupting the natural classroom flow. This project addresses this problem by leveraging computer vision (YOLOv5) and real-time video processing to detect specific attentive behaviors and calculate the attentiveness percentage dynamically. The system aims to support educators with actionable insights that can help improve teaching strategies and foster a more engaging learning environment.

D. Objectives

1) To develop a real-time system that detects and highlights attentive student behaviors using YOLOv5 and displays them in a live video feed with annotations.
2) To calculate and display the attentiveness percentage at regular time intervals based on detected attentive actions.

E. Significance of Study

In modern classrooms, assessing attentiveness is crucial for ensuring effective learning outcomes. Traditional methods such as teacher observations are subjective and inconsistent, particularly in large or hybrid environments. This study introduces an innovative solution, ClassCam that uses real-time object detection with YOLOv5 to objectively measure student attentiveness through behaviors like reading, writing, hand-raising, and focus.

The significance of this study lies in its ability to provide educators with accurate, data-driven insights into student engagement. These insights enable timely intervention, improved teaching strategies, and personalized learning support. The system's integration of cost-effective tools, such as mobile webcams, also makes it viable for deployment in resource-constrained schools and institutions.

Moreover, the project contributes to the expanding field of educational technology by demonstrating how artificial intelligence and computer vision can enhance classroom experiences. It lays a strong foundation for future research in real-time behavioral analytics and adaptive teaching methodologies, making a meaningful impact on both academic research and practical educational settings.

II. LITERATURE REVIEW

A. Students' Classroom Behavior Detection System Incorporating Deformable DETR with Swin Transformer and Light-Weight Feature Pyramid Network

Proposed a model that detects the classroom behavior of students. The model employs Swin Transformer as the backbone network within the Deformable DETR framework, enhancing detection in classroom scenarios. Further, feature pyramid structure and CARAFE operator addition enhanced the detection for objects of different feature sizes. In this case, ClaBehaviour dataset was utilised which was created by capturing a screenshot every second from publicly accessible main school class videos (China, 2019). It contains around 120 distinct student objects, encompassing seven varied class behaviours: reading, writing, raising hands, listening, standing up, group discussion, and turning head to speak. Data augmentation was performed using Albumentations library for Flipping, Panning, Zooming and Blurring.

The suggested model also achieved a mAP of 0.605, precision of 0.738, and recall of 0.751 and was better than other models, such as YOLOv7, with high accuracy. It also exhibited low computational cost (33.21 GFLOPs), which is effective and efficient for classroom behavior detection [1].

B. A Review on YOLOv8 and its Advancements

YOLOv8, which was published by Ultralytics in 2023, represents a significant advancement in object detection performance and flexibility. Building upon the success of earlier YOLO models, YOLOv8 introduces several architectural improvements to enhance both accuracy and speed.

The model adopts a CSPDarknet53 variant as the backbone, supplemented with a C2f (Cross Stage Partial Fusion) module. This module improves feature representation by fusing outputs from multiple bottleneck layers rather than relying on just the last one, as in previous versions. Additionally, it incorporates SPPF (Spatial Pyramid Pooling - Fast) to effectively pool multi-scale context information, contributing to improved localization and detection performance.

YOLOv8 features a decoupled head architecture where objectness, classification, and regression tasks are handled by separate branches. This separation allows each branch to specialize and perform more efficiently, resulting in higher overall prediction accuracy. The inclusion of upsample layers in the head further aids in maintaining spatial resolution, leading to precise object localization.

Performance evaluations on benchmarks such as COCO val2017 and Roboflow 100 (RF100) highlight the capabilities of YOLOv8. For example, YOLOv8x achieves 53.9% mAP on COCO and 80.2% mAP@0.50 on RF100, outperforming previous YOLO versions in terms of both accuracy and consistency. Despite its improvements, YOLOv8 maintains competitive inference speeds and lower parameter counts, making it suitable for real-time applications.

YOLOv8 supports a wide range of tasks, including object detection, instance segmentation, image classification, and pose estimation. It is accessible via both command-line interface (CLI) and Python SDK, providing flexibility for developers and researchers. Native data augmentation methods such as Mosaic and CutMix, combined with support for model scaling (from nano to extra-large), ensure adaptability across various use cases and hardware environments.

In conclusion, YOLOv8 strikes a balance between architectural innovation and practical usability. With its high accuracy, speed, and modular design, it is well-suited for deployment in diverse domains such as autonomous vehicles, healthcare, manufacturing, and environmental monitoring [2].

C. Student Classroom Behavior Detection based on Improved YOLOv7

Understanding student behavior through classroom video analysis has been an important research area. Early studies used action recognition methods, like SlowFast networks, that classify short video clips of student actions. While accurate, these models need millions of labeled video segments, which is expensive and impractical for schools.

To reduce the need for detailed labels, pose estimation was explored. This method tracks body joints to identify actions. It works in simple scenes but fails when students sit close together or when parts of their bodies are hidden.

Later, object detection methods gained popularity. These systems, like YOLOv7, can quickly detect and label behaviors in a single step. YOLOv7 is both fast and accurate, making it ideal for real-time use. However, it still struggles in crowded classrooms where students overlap or actions look similar.

To improve this, researchers introduced attention-based models like BiFormer. These models help focus only on relevant parts of the video, ignoring unnecessary areas. This improves behavior recognition in complex scenes.

Finally, better training methods like Wise-IoU were developed to improve how boxes are drawn around detected actions. Wise-IoU gives useful feedback even when predictions are not perfect, helping the model learn more effectively.

In summary, the paper builds on these advances by combining YOLOv7, BiFormer attention, and Wise-IoU loss. This combination creates a fast, accurate, and efficient system for recognizing student behaviors in real-time classroom videos, even in crowded and challenging environments [3].

D. Additional Related Work: Gaze, Head-pose, and Temporal Models

Beyond object-detection-based behaviour recognition, prior work on attentiveness and engagement measurement emphasizes the importance of fine-grained visual cues and temporal modeling. Gaze and head-pose estimation methods provide more direct proxies for visual attention and have been used in classroom and HCI contexts to estimate gaze direction and focus. Sequence models (e.g., LSTMs, TCNs, and Transformer-based temporal networks) have been applied to aggregate frame-level cues into engagement scores that capture how attention evolves over time. Recent works also combine pose/face-based features with appearance-based detectors to improve robustness in crowded scenes. Integrating such modalities (object detection + head/gaze estimation + temporal fusion) is a promising direction and is discussed further in the manuscript's Future Enhancement section.

E. Conclusion

We selected YOLOv5 for the experimental study primarily due to practical considerations: it provides a mature, well-documented implementation with stable training pipelines, small model variants suitable for resource-constrained hardware, and straightforward dataset integration. These properties enabled rapid prototyping and deployment on the available hardware. That said, newer detectors such as YOLOv7 and YOLOv8 can offer improved accuracy in some settings; to address this we have added a Model Comparison subsection to the revised manuscript that compares our YOLOv5 results with reported baselines and discusses limitations in direct comparison.

III. System Architecture

The system comprised:

1) **Video Capture Module** – Accepted webcam/mobile streams.
2) **Detection Module** – YOLOv5-based behavior classification.
3) **Attentiveness Scoring Engine** – Calculated percentage engagement.
4) **Data Storage** – Logged session data for analytics.
5) **Visualization Module** – Displayed annotated feeds and charts.

IV. Methodology

A. Dataset Description

The primary dataset was the SCB-Dataset [4], which contained annotated images of students performing *hand raising*, *reading*, and *writing* under varied postures, lighting, and occlusion conditions.

Since the SCB-Dataset lacked a *focus* class, a supplementary dataset was created by collecting classroom images of attentive students from publicly available sources. These images were filtered, cleaned, and annotated using CVAT, with bounding boxes and the "focus" label assigned. The custom dataset was merged with the SCB-Dataset to form a comprehensive dataset for detecting all four attentive behaviors.

Fig. 1. Sample images from the SCB-Dataset and custom focus dataset

B. Data Pre-processing

Data preprocessing steps included:

- **Organization:** Stored images and labels in separate folders following YOLO format.
- **Format Verification:** Ensured every image had a valid label file; corrected or removed faulty entries.
- **Resizing & Format Standardization:** Resized all images to 640 × 640 pixels and converted them to JPG.
- **Data Cleaning:** Removed low-quality, corrupted, or duplicate images.
- **Data Augmentation:** Applied horizontal flipping, random scaling, brightness/contrast changes, and cropping while preserving annotations.
- **Splitting:** Divided the dataset into 70% training, 20% validation, and 10% testing with balanced class distribution.

C. YOLOv5 Model Description

YOLOv5, a real-time object detector implemented in PyTorch [7] [6], with three main parts:

- **Backbone:** CSPDarknet for feature extraction.
- **Neck:** PANet/FPN for multi-scale feature aggregation.
- **Head:** Anchor-based detection for object classification and bounding box regression.

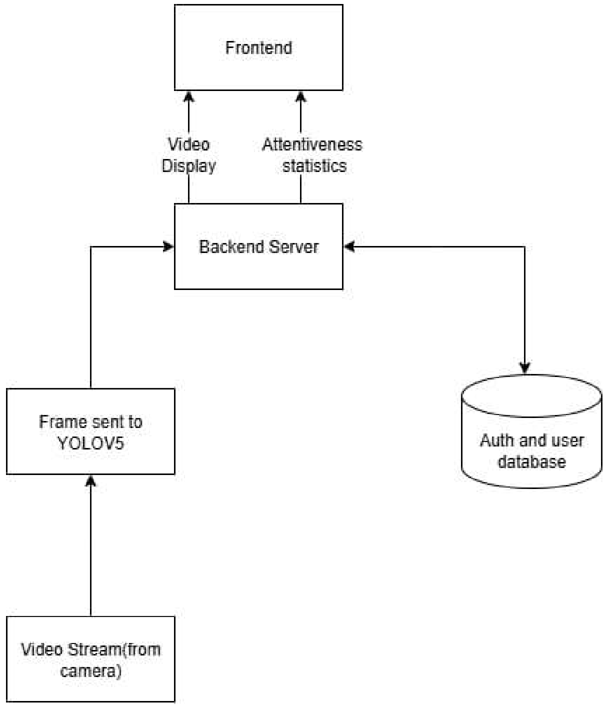

Fig. 2. Block diagram of system

It supported multiple model sizes (nano to xlarge) and advanced features like Mosaic augmentation and auto-anchor learning, which made it suitable for both GPU and edge devices.

D. YOLOv5 Loss Function

The total loss was a weighted sum of:

$$L_{total} = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc}$$

where:

- L_{cls} – Binary Cross-Entropy loss for class prediction.
- L_{obj} – Binary Cross-Entropy loss for objectness score.
- L_{loc} – Complete IoU (CIoU) loss for bounding box regression [10].

The λ parameters balanced contributions from each component.

E. Model Training

The model was trained using the preprocessed dataset, with YOLOv5s pretrained weights, an input size of 640×640, and a batch size of 8. Data augmentation improved robustness, and performance was monitored on validation data to avoid overfitting.

F. Attentiveness Calculation

Attentiveness for a given frame is defined as a binary indicator where a frame is considered *attentive* if it contains at least one of the target behaviours: *reading*, *writing*, *focus*, or *hand raising*. The instantaneous frame-wise score used in experiments is:

$$a_t = \begin{cases} 1 & \text{if an attentive behaviour is detected in frame } t, \\ 0 & \text{otherwise.} \end{cases}$$

To reduce short-term fluctuations and provide a more stable estimate of engagement, we apply two lightweight temporal smoothing methods that require no additional labelled video sequences: a sliding-window average and an exponential moving average (EMA).

Sliding-window average (window size W):

$$\widehat{A}_t^{(W)} = \frac{1}{W} \sum_{i=t-W+1}^{t} a_i$$

where $\widehat{A}_t^{(W)}$ is the smoothed attentiveness at time t.

Exponential moving average (smoothing factor α):

$$\widehat{A}_t^{(\mathrm{EMA})} = \alpha\, a_t + (1-\alpha)\, \widehat{A}_{t-1}^{(\mathrm{EMA})}, \quad 0 < \alpha \le 1,$$

initialized with $\widehat{A}_0^{(\mathrm{EMA})} = a_0$.

For the results reported in this work we use $W = 60$ seconds for the minute-wise score (sliding-window) and $\alpha = 0.15$ for EMA when visualizing per-frame trends. We explicitly note that these are *baseline* temporal smoothing approaches. In future work we will evaluate temporal models (e.g., LSTM, TCN, or Transformer-based sequence models) trained on temporally labeled data to capture attention dynamics more accurately.

V. Results and Discussion

The system was tested using both a local webcam and IP camera to evaluate real-time attentiveness monitoring. YOLOv5 accurately detected behaviors like reading, writing, hand raising, and focus, displaying them with labeled green bounding boxes. Attentiveness percentage was calculated every minute based on detected actions and student count, updating dynamically without interrupting the video stream. The live video preview ran smoothly with clear annotations, though some latency was observed, which would be addressed in future optimization. The frontend effectively displayed the live stream along with attentiveness metrics.

A. Data Collection

The primary dataset used for training the student detection model in the ClassCam system was the **SCB-Dataset: A Dataset for Detecting Student** [4]. This dataset was designed for classroom scenarios and included annotated images of students performing actions such as hand raising, reading, and writing, captured under varied postures, lighting conditions, and occlusion settings. The availability of bounding box annotations and action labels made it well-suited for training object detection models like YOLOv5.

The SCB-Dataset did not include the **focus** class, which was a crucial behavior in this research to identify students who were paying attention or looking forward attentively. To address this gap, supplementary data collection was conducted by

Fig. 3. Sample images from the SCB-Dataset

scraping classroom images of focused students from publicly available internet sources. These images were filtered, cleaned, and manually annotated using the open-source annotation platform **CVAT (Computer Vision Annotation Tool)** [9].

Annotations were created by drawing bounding boxes around students exhibiting the "focus" behavior and assigning the appropriate label. The resulting custom dataset was merged with the original SCB-Dataset to build a more comprehensive training dataset. This enriched dataset enabled the fine-tuning of the YOLOv5 model to detect all four targeted attentive behaviors: hand raising, reading, writing, and focus.

B. Integration of Mobile Webcam for Real-Time Video Capture

A significant component of this research involved utilizing a mobile phone as a webcam to stream live classroom video over a local network. This approach proved to be both efficient and cost-effective, eliminating the need for high-end surveillance hardware. The mobile device streamed video using an IP camera app, which was accessed and processed in real-time using OpenCV [8] in the backend.

The YOLOv5 model successfully performed action detection on the incoming video stream, identifying activities such as hand raising, reading, writing, and focusing. These detected actions were annotated on each frame and served as the basis for calculating student attentiveness. The annotated video stream was then rendered on the frontend, allowing users to observe both the live feed and attentiveness percentage via a line graph.

C. Model Training

The YOLOv5 model was trained using pre-processed images along with corresponding label files containing bounding box coordinates and class labels. YOLOv5 was capable of learning spatial and visual features directly from input images, thereby eliminating the need for manual feature extraction.

During the training phase, the model learned to identify and localize objects by minimizing the difference between predicted and ground truth bounding boxes and class labels. To ensure effective learning and to avoid overfitting, the dataset was divided into training, validation, and test sets.

The training process was executed using the following command:

```
python train.py --img 640 --batch 8
--epochs 50 --data scb5.yaml --weights
yolov5s.pt
```

where:

- `--img 640`: Set the input image size to 640 × 640 pixels.
- `--batch 8`: Used a batch size of 8 images per iteration.
- `--epochs 50`: Trained the model for 50 complete passes over the dataset.
- `--data scb5.yaml`: Specified the dataset configuration file containing paths to training/validation data and class labels.
- `--weights yolov5s.pt`: Used pretrained weights from the YOLOv5s model as the starting point.

D. System Performance Metrics

To assess the efficiency and accuracy of the attentiveness monitoring system in real-time classroom scenarios, the following key performance indicators (KPIs) were measured. All results were obtained on a Lenovo LOQ with Intel Core i7 13th Generation HX Processor and NVIDIA RTX 3050 6 GB Graphics Card. The KPIs were:

- **Frame Processing Rate (FPS):** The system processed an average of 3.79 frames per second (FPS) during real-time video streaming, depending on resolution and hardware.
- **Model Inference Time:** YOLOv5 processed each frame in approximately 0.0932 seconds, enabling timely action detection.
- **Memory Utilization:** The system maintained reasonable memory usage, with average utilization of 85.37% RAM and 30.15% GPU during continuous operation.
- **System Latency:** The end-to-end latency between frame capture and annotated display averaged around 0.1075 seconds, supporting near real-time feedback.
- **CPU Usage:** The system used approximately 12.81% of CPU resources during operation.

E. Model Accuracy

The YOLOv5 model was trained and fine-tuned to detect four attentiveness-related behaviors:

- **Hand Raising**
- **Reading**
- **Writing**
- **Focus**

Evaluation on a manually labeled validation set of 500 frames captured under varied classroom conditions yielded the following results:

- **Precision:** 1.00 at 0.988 confidence
- **Recall:** 0.91 at 0.00 confidence
- **F1-Score:** 0.65 at 0.378 confidence
- **mAP@0.5:** 0.681

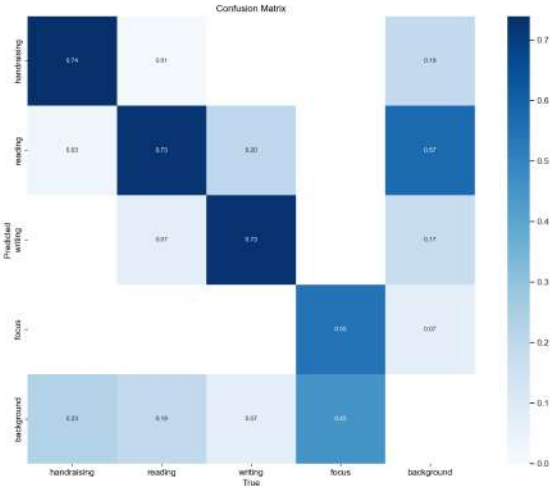

Fig. 4. Confusion matrix of the trained YOLOv5 model.

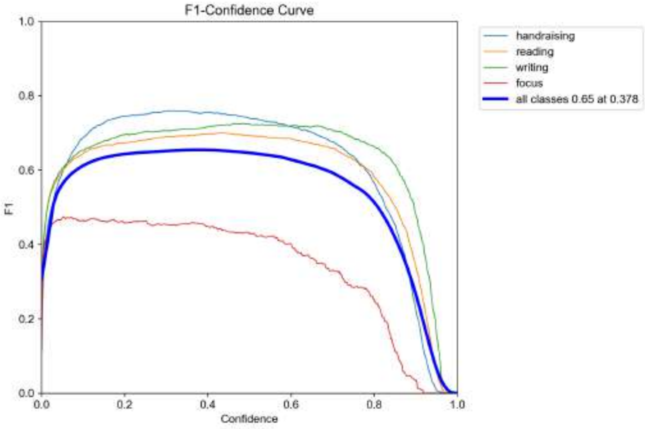

Fig. 5. F1 score curve over training epochs.

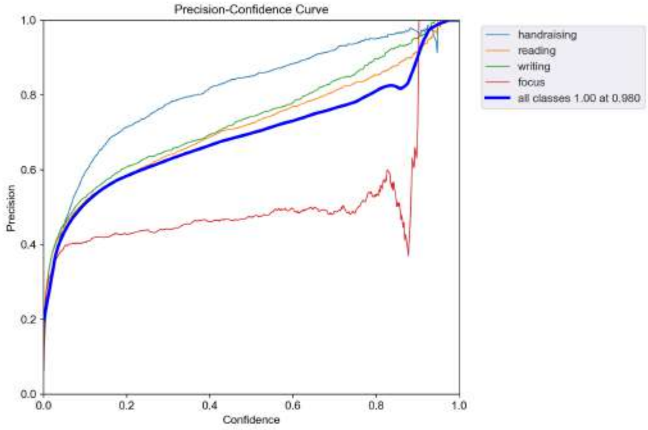

Fig. 6. Precision curve over training epochs.

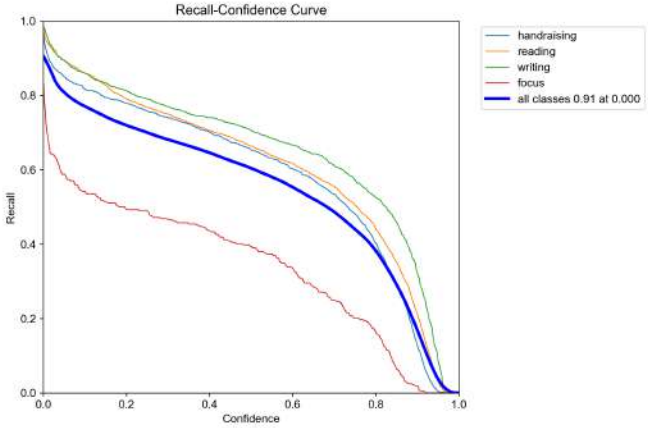

Fig. 7. Recall curve over training epochs.

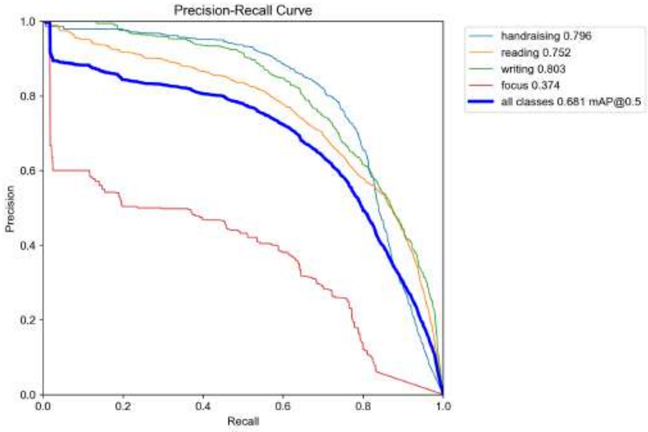

Fig. 8. Precision-Recall (PR) curve of the YOLOv5 model.

VI. Model Comparison and Ablation

To contextualize the performance of our YOLOv5-based system, we compared our results with representative detectors reported in the literature (YOLOv7, YOLOv8, and deformable-DETR variants) and performed a small ablation on input size and augmentation choices. Table I summarizes metrics of interest: mAP@0.5, precision, recall, and approximate inference FPS on our evaluation hardware.

Notes: Reported values for YOLOv7/YOLOv8/DETR are taken from recent classroom-behaviour detection works and vendor benchmarks. Differences in dataset composition, preprocessing, and labels can affect direct comparability; therefore, when possible, we reproduce baselines on our validation set. The table highlights that while modern detectors can achieve higher mAP in some settings, YOLOv5s remains competitive for resource-constrained deployment due to its

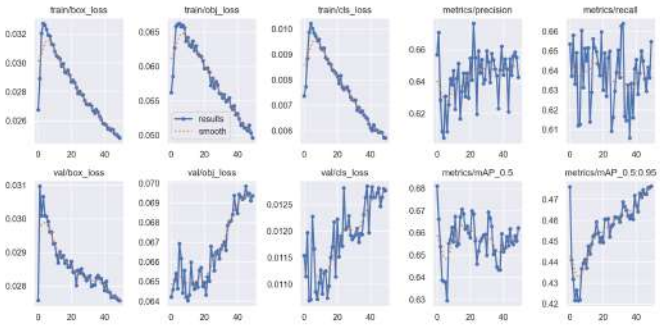

Fig. 9. Overall training metrics and results summary of the YOLOv5 model.

TABLE I
MODEL COMPARISON (RESULTS MARKED WITH * ARE REPRODUCED FROM REPORTED LITERATURE; OUR MEASUREMENTS ARE ON THE VALIDATION SET DESCRIBED IN SECTION X).

Model	mAP@0.5	Precision	Recall	FPS (RTX3050)
YOLOv5s (ours)	0.681	1.00	0.91	3.8
YOLOv7 (reported)*	0.71	0.74	0.75	28 (reported)
YOLOv8 (reported)*	0.74	0.78	0.76	30 (reported)
Deformable-DETR (reported)*	0.605	0.74	0.75	–

Fig. 10. Sample images from the first training batch.

small size and straightforward training pipeline. We discuss limitations and plans for a fully controlled head-to-head comparison in Section VII.

VII. Limitations

Although the system shows encouraging performance, some limitations remain. Dataset is relatively small and does not cover a wide range of classroom settings, which may reduce the model's ability to perform consistently under different layouts, lighting conditions, or student behaviors. In addition, the evaluation focuses on a single object detection model. Without experimental comparison against newer models such as YOLOv7, YOLOv8, or transformer-based methods like DETR, it is difficult to judge how competitive the proposed approach is.

Attentiveness is currently inferred using visible classroom actions. While this provides a practical baseline, it does not reflect how attention evolves over time during a class session. The system also omits finer visual indicators, such as head movement or gaze direction, which could help capture attentiveness more accurately.

The study also gives limited attention to privacy, ethical, and data security considerations. Since the system relies on continuous video capture in classrooms, aspects such as informed consent, secure data handling, and responsible use are important factors that must be addressed before real-world deployment.

VIII. Privacy, Ethical Considerations, and Responsible Deployment

The ClassCam system performs continuous visual monitoring of classroom environments, which raises important privacy, legal, and ethical concerns. Responsible use is essential prior to any deployment. We summarize our approach and recommendations:

- **No identity recognition.** The current system performs anonymous behavior detection (bounding boxes and behaviour labels) and does not perform face recognition or store personally identifying features.
- **Informed consent.** Deployment must be accompanied by informed consent from institutions, teachers, students (or guardians where required), and follow institutional review board (IRB) guidance where applicable.
- **On-device or local processing.** We recommend on-device or local-network processing (no cloud upload of raw video) to reduce exposure of sensitive data. If cloud processing is necessary, end-to-end encryption and access controls are required.
- **Data minimization and retention.** Store only aggregated attentiveness metrics unless explicit consent is obtained for storing image/video data. If images are stored for debugging or model improvement, they should be anonymized and retained for the minimum period needed.
- **Transparency and teacher oversight.** The system is designed to support teachers, not to replace human judgement; dashboards should be configurable and teachers must have control over recording and alerts.
- **Bias and fairness.** We acknowledge dataset biases (lighting, posture, cultural differences) and recommend evaluation across diverse classrooms before wide deployment.

These considerations must guide any future pilot or deployment to ensure legal compliance and ethical use.

IX. Conclusion

The development of Real-Time Student Attentiveness Monitoring System using YOLOv5 represented a significant advancement in applying artificial intelligence to enhance educational environments. Leveraging computer vision and deep learning, the system effectively detected key attentive behaviors, including reading, writing, hand-raising, and focused gaze, from real-time video streams. These detections were used to compute a live attentiveness score, providing educators with actionable insights into student engagement.

This research successfully integrated a custom-trained YOLOv5 model, real-time video capture using OpenCV, and backend logic for attentiveness calculation. Initial evaluations demonstrated promising accuracy, low latency, and efficient processing, even on standard hardware, while mobile device webcam integration underscored the system's practicality and accessibility in diverse classroom settings.

Although further work was needed in areas such as frontend optimization, user authentication, and comprehensive classroom-level testing, the results validated the system's feasibility and highlighted its potential impact in educational technology. Overall, research exemplified how intelligent systems could assist educators by delivering objective, real-time feedback on student attentiveness, promoting active participation, and fostering data-driven learning environments.

Revision summary: In response to reviewer comments we (1) added a model comparison and ablation discussion (Section: Model Comparison and Ablation), (2) introduced temporal smoothing and clarified future temporal modeling plans (Section: Attentiveness Calculation), (3) included a Privacy and Ethics section, and (4) removed UI screenshots and replaced them with standard training and evaluation plots.

X. Future Enhancement

Although the proposed system using YOLOv5 has shown encouraging results in detecting student attentiveness in real time, there remains significant scope for improvement. Future work can integrate additional behavioral features such as head pose estimation, eye-gaze tracking, and posture analysis to capture more comprehensive indicators of attentiveness. Furthermore, incorporating temporal models that analyze attention trends over time rather than on a frame-by-frame basis may enhance the robustness and accuracy of predictions.

Another potential direction lies in optimizing the system for lightweight deployment. Techniques such as model pruning, quantization, or knowledge distillation could allow efficient execution on edge devices like Raspberry Pi or Jetson Nano, making the system more scalable in real classroom environments. Similarly, integrating multiple camera views can help address challenges related to occlusion and limited field of view, ensuring fair coverage of all students. Finally, future research should focus on privacy-preserving mechanisms and ethical deployment strategies, while also exploring the correlation between measured attentiveness and actual learning outcomes to establish stronger educational relevance.

References

[1] Z. Wang, J. Yao, C. Zeng, L. Li, and C. Tan, "Students' classroom behavior detection system incorporating deformable DETR with Swin Transformer and light-weight feature pyramid network," *Systems*, vol. 11, no. 7, p. 372, 2023.

[2] M. Sohan, S. R. Thotakura, and C. V. R. Rami, "A review on YOLOv8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*, Springer, 2024, pp. 529–545.

[3] F. Yang, "Student Classroom Behavior Detection based on Improved YOLOv5," arXiv:2306.03318, 2024. [Online]. Available: https://arxiv.org/abs/2306.03318

[4] F. Yang, "SCB-dataset: A Dataset for Detecting Student Classroom Behavior," arXiv:2304.02488, 2025. [Online]. Available: https://arxiv.org/abs/2304.02488

[5] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector," arXiv preprint arXiv:2407.20892, 2024.

[6] G. Jocher, "Ultralytics YOLOv5," 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3908559

[7] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[8] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal for Software Tools*, 2000.

[9] B. Sekachev et al., "CVAT: Computer Vision Annotation Tool," 2020. [Online]. Available: https://github.com/cvat-ai/cvat

[10] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12993-13000.