Deep Learning-based Student Learning Behavior Understanding Framework in Real Classroom Scene

Yuxin Yang[†], Zhengyong Ren[†], Christian Lenart[†], Ashton Corsello[†], Karl Kosko[†], Simon Su[△], Qiang Guan[†]

†: Kent State University, USA
△: National Institute of Standards and Technology, USA
{yyang45, zren2, clenart4, acorsell, kkosko1, qguan}@kent.edu
{simon.su}@nist.gov

***Abstract*—Deep learning techniques have emerged as valuable tools for video analysis and motion detection. Recent advancements in this field have shown promising results. Our objective is to leverage these video understanding techniques to aid teachers in evaluating their teaching quality and enhancing their effectiveness in the classroom. However, existing research on student behavior analysis primarily focuses on recognizing actions pertaining to classroom management, neglecting the identification of "learning behaviors" exhibited by students. To address this limitation, we introduce a novel video dataset specifically designed to capture the nuances of "learning behaviors" displayed by primary-grade students in the mathematics classroom, along with a dedicated student localization dataset focused on detecting the location of individuals. Our approach introduces a framework that utilizes deep learning-based object detection and action recognition techniques trained on our curated datasets to analyze and comprehend student learning behaviors in the classroom. To assess the performance of our approach, we conduct separate tests on our object detection and action recognition models. Subsequently, our framework is applied to a collection of recorded 360-degree classroom videos, enabling a thorough evaluation of its capabilities.**

***Index Terms*—deep learning, action recognition, object detection, student learning behavior analysis**

I. Introduction

Analyzing and understanding student behavior plays a pivotal role in elevating the quality and effectiveness of teaching within the classroom environment. To support teachers in their professional development, classroom videos have become widely adopted as primary training materials [1]. These videos serve as valuable resources that enable educators to observe and grasp student behavior comprehensively. However, traditional methods of analyzing classroom videos are often laborious and time-consuming, demanding heavy attention from teachers as they carefully watch each video frame by frame to identify individual students and interpret their behaviors accurately. Fortunately, recent advancements in deep learning techniques in the field of computer vision have introduced a promising solution to expedite and enhance the analysis of student behavior within classroom videos. The utilization of deep learning techniques in analyzing student behavior brings numerous benefits, including improved efficiency and scalability. Instead of relying solely on human observation, teachers can now rely on computer algorithms to detect and classify specific student actions, such as counting numbers or manipulating blocks. This automated analysis significantly reduces the time and effort required from teachers, enabling them to allocate more energy towards developing personalized instructional strategies and addressing individual student needs.

Many existing studies on student behavior understanding utilizing deep learning techniques focus on either the analysis of facial features or the recognition of student actions. Facial feature analysis, such as facial expressions or viewpoints, only offers a limited understanding of students in a classroom scene. For instance, in [2], a simple convolutional neural network (CNN) is developed to determine a student's comprehension based on facial expressions. Other works, such as [3], propose combinations of facial expressions, and input data from keyboard and mouse, but primarily study the student's engagement level. Additionally, [4] combines facial viewpoints with recorded teacher speech audio to analyze student attention and teaching style. On the other hand, action analysis provides a more comprehensive understanding of students in the classroom scene. Some studies, like [5], utilize skeleton data to recognize student actions, while others, such as [6], employ a combination of 3D CNN and Long Short-Term Memories (LSTM) for action recognition, albeit with students facing away from the cameras, which can compromise the effectiveness of the method. Another approach [7] combines 2D CNN for spatial information and 1D CNN for temporal information to recognize student actions, yet the proposed dataset remains relatively small. Although these individual works have provided valuable insights into several classroom management aspects, such as recognizing actions of raising hands and measuring engagement levels, there is a critical need to focus on identifying specific instances where students engage in "learning behaviors". In the context of primary-grade mathematics learning, these learning behaviors encompass activities like counting, using manipulatives, and writing. Addressing this need for a deeper understanding of student learning behaviors in the classroom is essential for developing more effective educational interventions and improving the overall teaching quality and learning experience.

To overcome these limitations, we take a proactive approach by curating a comprehensive video dataset specifically captur-

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00067

ing students engaged in classroom-specific learning behavior actions. Furthermore, we propose a robust framework for understanding student behavior, employing deep learning techniques to accomplish two key tasks: student localization and comprehensive analysis of their actions within the classroom environment. Additionally, we record a dedicated set of 360-degree videos in the authentic classroom setting, allowing us to validate the effectiveness and practicality of our proposed framework. Through these efforts, we aim to overcome existing limitations and advance the understanding of student behavior in real-world classroom scenarios.

In Section II, we review commonly used techniques on object detection and action recognition. We outline the details of our proposed framework and the datasets used to develop it in Section III. We provide the numerical tests and results in Section IV. Finally, we offer further discussion on our findings, future works, and conclusions in Section V.

II. Related Works

We will first delve into commonly employed techniques for object detection, then we will offer valuable insights into action recognition techniques.

A. Object Detection

Object detection is a technique used to identify the location and category of objects in an image. Traditional object detection methods rely on hand-crafted features to locate objects, such as Local Binary Patterns (LBP) [8] and Scale Invariant Feature Transforms (SIFT) [9]. However, these methods perform poorly and are not easily transferable to other scenarios, making them unsuitable for our problem.

Recent advancements in deep learning techniques have made deep learning the preferred method for developing object detection algorithms [10]. For instance, You Only Look Once (YOLO) [11], Single Shot Detector (SSD) [12], and RetinaNet [13] are region-free object detection methods that use regression to generate bounding boxes. While region-based methods [14] involve generating object region proposals, extracting features from the region proposals, and predicting the object class in each region proposal.

Region-based Convolutional Neural Network (R-CNN) [15] integrates convolutional neural networks with region-based methods to improve the performance of region-based object detection. Fast R-CNN [16] has faster detection speed and better prediction accuracy than R-CNN. Faster R-CNN [17] develops a region proposal network (RPN) that generates object location proposals more efficiently and accurately. Mask R-CNN [18] adds a parallel object mask to Faster R-CNN, improving both speed and performance.

B. Action Recognition

The field of action recognition has seen rapid progress in recent years, thanks to the rapid development of deep learning techniques. There are two main categories of deep learning-based action recognition models: vision-based and skeleton-based. Vision-based techniques typically use convolutional neural networks (CNNs) to directly learn and recognize actions from video frames [19]. More recently, vision transformers have been proposed [20] and have led to the development of new techniques such as multiscale vision transformer [21] and video swin transformer [22]. Skeleton-based techniques, on the other hand, use human skeleton data as input instead of raw video. Because skeleton data has a smaller data size, skeleton-based techniques have lower computational costs compared to vision-based techniques. Skeleton data can be treated as graph data with nodes and edges, and several skeleton-based action recognition techniques, such as ST-GCN [23], use Graph Neural Networks (GNNs) [24] to recognize human actions from skeleton data. However, it can be challenging to distinguish between similar actions in classroom environments, where students mainly move their hands, such as writing and manipulating math manipulatives. Moreover, skeleton-based action recognition requires preprocessing raw videos to obtain human skeleton data using tools like OpenPose [25].

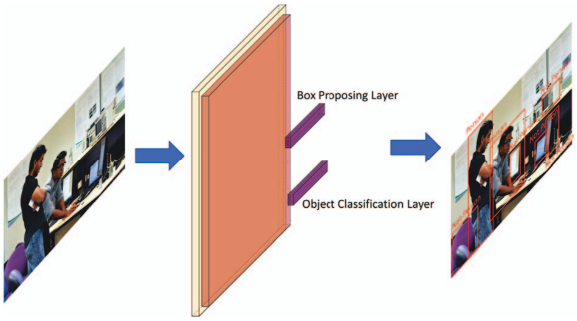


Fig. 1: The schematic illustration of our student localization model.

All of these works provide valuable insight into how we can design a deep learning-based framework that combines both object detection techniques and action recognition techniques to understand students' behaviors in the classroom.

In the next section, we will discuss the details of our approach, including essential aspects of the student localization method, the action recognition method, and a comprehensive presentation of the design details behind our proposed student learning behavior understanding framework.

III. Methodology

Our proposed student learning behavior understanding framework consists of two parts: a student localization model that identifies students and provides bounding boxes around them in the video, and an action recognition model that recognizes the behaviors of the detected students.

A. Student Localization

We first discuss the technical details of the student localization model, which is based on the Region Proposal Network (RPN) in Faster R-CNN [17].

The student localization model, illustrated in Fig. 1, begins by using a CNN, denoted as $\mathcal{G}$ to extract image features, F,

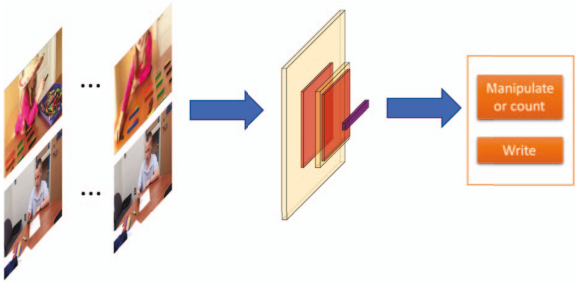


Fig. 2: The schematic illustration of our action recognition model.

which will be used as input of RPN. The RPN, denoted as $\mathcal{F}$, comprised of a convolution layer with an $n \times n$ square convolutional kernel, proposes k different bounding boxes. This convolutional layer is fully connected to a middle linear layer with m-dimension, which is then connected to two final linear layers. The first final layer is the box proposing layer, used to produce parameters for each box, while the second final layer is the object classification layer, used to predict whether or not there is an object in each box.

The dimension of the box proposing layer is $4k$, as each proposed box has four parameters: the x and y coordinates for the center point, as well as the height (h) and width (w) of the box. The object classification layer applies a sigmoid function $\sigma(x)$ to limit the output ranging from 0 to 1, where $sigma(x) = \frac{1}{1+e^{-x}}$. When there is an object in the box, the output of the object classification layer is used to determine whether the detected object is a person, with $\sigma(x) \geq 0.5$ indicating a person, and $\sigma(x) < 0.5$ indicating that there is no person in the box. The dimension of the object classification layer is k.

For our work, we followed the original paper and set $n = 3$ and $k = 9$. To extract image features, we used ResNet-50 [26] pretrained on the ImageNet dataset [27] instead of VGG-19 [28]. This is because ResNet-50 has shown better performance than VGG-19 in many computer vision tasks.

B. Student Action Recognition

We will now delve into the technical details of the student action recognition model, which is based on the Multi-Head Pooling Attention (MHPA) in Multiscale Vision Transformer (MViT) [21]. As shown in Fig. 2, we start by processing the input video clip x, which is a tensor of shape $T \times H \times W \times C$, where T is the number of video frames, H and W are the height and width of the video, and C is the number of channels. We flatten x and split it into N non-overlapping tensor tiles, denoted by $x_p \in \mathbb{R}^{T \times P \times P}$, where P is the size of the split tensor tile and $N = HWC/P^2$ is the number of tiles. We then apply a linear layer to process each tensor tile into a latent vector with a latent dimension of D, resulting in the tensor x_D. Next, we encode the position information of x_D using positional embedding.

The transformer block in our action recognition model consists of three components: an MHPA, a Multi-Layer Perceptron (MLP), and layer normalization (LN) [29]. Each transformer block performs the following operations:

$$\begin{aligned} x_{D,1} &= \text{MHPA}(\text{LN}(x_D)) + x_D, \\ x_{D,2} &= \text{MLP}(\text{LN}(x_{D,1})) + x_{D,1}, \end{aligned} \tag{1}$$

where $x_{D,1}$ is the output of the MHPA and $x_{D,2}$ is the output of the transformer block. The MHPA applies a multi-head self-attention mechanism, while the MLP is a feedforward neural network with a single hidden layer. Layer normalization is applied before and after the MHPA and MLP to normalize the output of each layer.

The MHPA comprises multiple attention modules that compute the query Q, key K, and value V for the input tensor x_D as follows:

$$Q = x_D W_Q, \; K = x_D W_K, \; V = x_D W_V, \tag{2}$$

where $W_Q \in \mathbb{R}^{D \times D}$, $W_K \in \mathbb{R}^{D \times D}$, and $W_V \in \mathbb{R}^{D \times D}$ are the matrices that convert the input x_D to the query, key, and value tensors. The resulting Q, K, and V tensors have the same dimension as x_D ($T \times P \times P \times D$). Subsequently, a pooling operation is applied to Q, K, and V to reduce their size along the first three dimensions. Finally, we compute the attention score of Q, K, and V using the softmax function as follows:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D}}\right) V. \tag{3}$$

To parallelize the attention computation, we use Multi Head Attention (MHA) [30]. With h heads in MHA, each head $head_i, i \in [1, h]$ has a unique set of matrices, namely $W_{Q,i} \in \mathbb{R}^{D \times D}$, $W_{K,i} \in \mathbb{R}^{D \times D}$, and $W_{V,i} \in \mathbb{R}^{D \times D}$, to compute the query $Q_i = x_D W_{Q,i}$, key $K_i = x_D W_{K,i}$, and value $V_i = x_D W_{V,i}$, respectively. Each head then computes the attention using Q_i, K_i, and V_i:

$$\begin{aligned} head_i(x_D) &= \text{Attention}(Q_i, K_i, V_i) \\ &= \text{Attention}(x_D W_{Q,i}, x_D W_{K,i}, x_D W_{V,i}). \end{aligned} \tag{4}$$

After computing the attention of each head $head_i$ using their respective query, key, and value matrices $W_{Q,i}$, $W_{K,i}$, and $W_{V,i}$, we concatenate the outputs from each head and apply a linear transformation with matrix $W_O \in \mathbb{R}^{hD \times D}$ to obtain the final output in $\mathbb{R}^{T \times P \times P \times D}$:

$$\text{MHA}(Q, K, V) = \text{Concat}(head_1, ..., head_h) W_O. \tag{5}$$

And we finally use an MLP combined with the Softmax to predict the probability of each action class:

$$\hat{y} = \text{Softmax}(\text{MLP}(\text{MHA}(Q, K, V))), \tag{6}$$

where $\hat{y} \in \mathbb{R}^a$ is the probability of each action class and a is the number of action classes.

We use Binary Cross Entropy (BCE) loss as the optimality criterion to compute the loss between the ground truth and the predicted probability:

$$\mathcal{L} = \frac{1}{N} \sum_n \left(\hat{y}_n \cdot \log(y_n) + (1 - \hat{y}_n) \cdot \log(1 - y_n) \right), \tag{7}$$

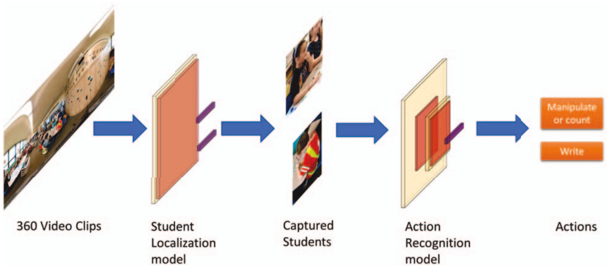


Fig. 3: Schematic illustration of our deep learning-based student behavior understanding framework.

where N is the batch size, $n = 1, 2, ..., N$, $\hat{y}_n$ is the predicted probability, and y_n is the ground truth.

C. Student Learning Behavior Understanding Framework

Here we discuss the design details of our deep learning-based student learning behavior understanding framework.

We show the schematic illustration of our proposed deep learning-based student learning behavior understanding framework in Fig. 3. Our framework comprises two stages. The first stage involves detecting students in each frame using our trained student localization model. In the second stage, neighboring frames are utilized to recognize the actions of each student through our trained student action recognition model. Subsequently, we report the understanding results in both video and text formats.

In the video format, a bounding box is provided for each student, along with the corresponding recognized action that has the highest probability. Similarly, in the text format, we provide the coordinates of the bounding box for each student, along with the recognized action and its corresponding probability.

The process of student learning behavior understanding is shown in Algorithm 1, which outlines the step-by-step procedure for utilizing our framework to analyze and comprehend student actions within a classroom environment.

D. Dataset

1) Student localization dataset: COCO dataset [31] is a widely used large-scale dataset for object detection, segmentation, and captioning tasks. The original COCO dataset contains 80 different object categories. As we aim to localize students in videos using our student localization model, we only keep 2 object categories, person and non-person where the images of our proposed person category are the images from the person category of the original dataset, and the images of our proposed non-person category are the union of images from all other categories in the original dataset. Since in the original COCO dataset, the number of non-person objects is more than twice the number of persons, we randomly remove half of the original non-person objects. Our proposed student localization dataset has a total of 585,124 annotated objects, where 273,468 are persons and 311,656 are non-persons.

Algorithm 1 Student Learning Behavior Understanding

Input: Classroom environment video V, Student localization model Loc, Student action recognition model Act, Number of frames used to recognize action z

Output: Location coordinates and actions R of each student in the video

```
Initialize: $T \leftarrow \emptyset$, $A \leftarrow \emptyset$
for each frame $f_v$ in $V$ do
    $C \leftarrow Loc(f_v)$
    for each detected object $o$ in $C$ do
        if object category of $o$ is non-person then
            $pass$
        else
            $t \leftarrow$ location coordinate of $o$
            $v \leftarrow z$ neighboring frames of $f_v$ within $t$
            $c \leftarrow Act(v)$
            $R \leftarrow R \cup \{(t, c)\}$
        end if
    end for
end for
return $R$
```

2) Student learning behavior dataset: In our endeavor to comprehend primary-grade student learning behaviors within the mathematics classroom environment, we focused on three major actions: manipulation of math manipulatives, counting objects, and writing.

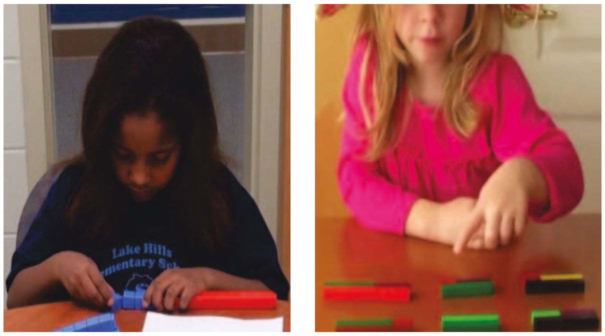
Our curated dataset includes 170 videos collected from Vimeo (https://vimeo.com/) featuring students performing at least one of these actions and recorded an additional 15 videos, each featuring only one of the three actions. We cropped all the collected videos to a size of 224×224 and split them into 5-second video clips, resulting in a total of 1,805 video clips.

After analyzing the video clips, we found that the actions of ‘manipulating’ and ‘counting’ were too similar, with students working on objects in both cases, as shown in Fig. 4. Therefore, we merged ‘manipulating’ and ‘counting’ into a single action in our dataset. The statistics of our final dataset are shown in Table. I.

We also created a series of pre-recorded 360-degree videos featuring classroom teaching environments to test our proposed student behavior understanding framework. These videos feature an original resolution of 5760×2880 and include 24 individuals, with three teachers and 21 students present in each video. The total duration of our 360-degree classroom videos is 33 minutes and 22 seconds. As shown in Fig. 5, the videos provide an immersive learning experience and allow viewers to explore the entire classroom environment.

Action	Manipulating and counting	Writing	Total
Number of video clips	1,189	616	1,805

TABLE I: Statistics of our student actions dataset.



(a) (b)

Fig. 4: (a) A student doing manipulation, and (b) a student doing counting.

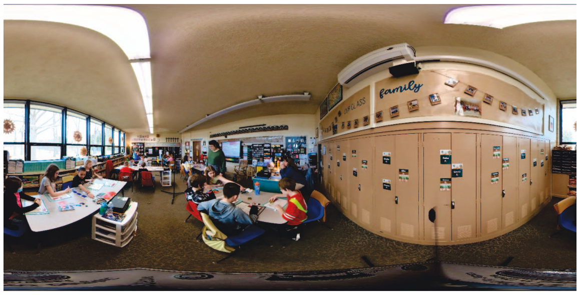


Fig. 5: One frame of our pre-recorded 360-degree classroom environment video.

IV. Experiments

To verify the efficacy of our approach in analyzing students' behavior within a classroom setting, we carried out three experiments: student localization, student action recognition, and video detection on the pre-recorded 360-degree classroom environment videos using our proposed deep learning-based student behavior understanding framework.

A. Experiment Setup

Our training of the student localization model builds upon the pretrained Faster R-CNN model, specifically trained on the original COCO dataset [31]. In this process, we maintain the fixed state of all layers in ResNet-50 fixed and solely finetune the Region Proposal Network (RPN) using our dedicated student localization dataset. The student localization model undergoes a rigorous training process spanning 60 epochs. For training, we utilize 96% data, amounting to 561,232 annotated objects, as our comprehensive training dataset. The remaining 4% of the data, comprising 23,892 annotated objects, is reserved as the test dataset to evaluate the model's performance. To optimize the student localization model, we employ Stochastic Gradient Descent (SGD) [32] with a learning rate of 1×10^{-4}. This choice of optimization method aids in refining the model's parameters and enhancing its accuracy in localizing students within the classroom environment.

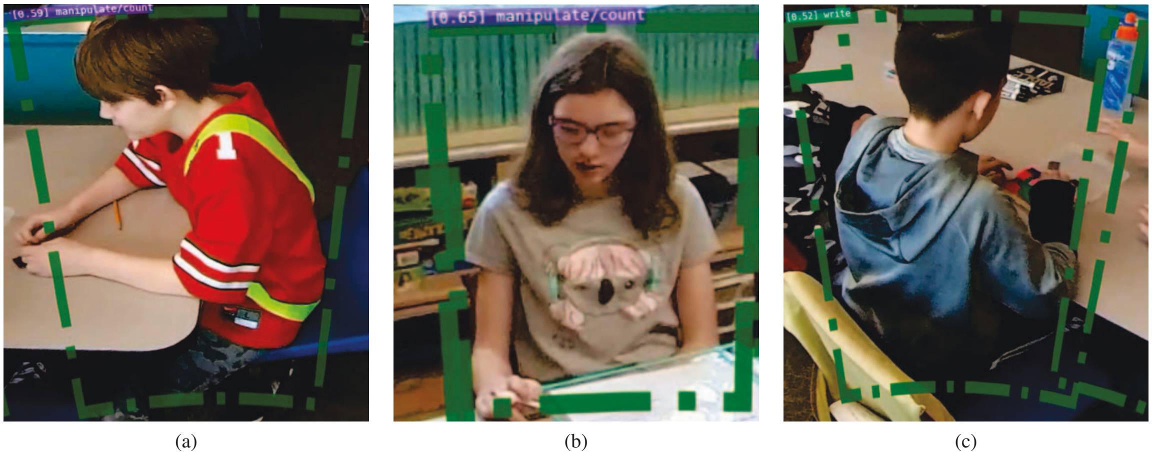
Our training process for the student action recognition model builds upon the pretrained MViT-B model [21], which was initially trained on the Kinetic-400 dataset [33]. To leverage the advantages of the pretrained model, we follow [34] to employ a two-step approach consisting of linear probing followed by finetuning. The model is trained for a total of 80 epochs. Specifically, we allocated 20 epochs for the linear probing stage and an additional 60 epochs for the finetuning stage. This multi-stage training approach ensures that the model effectively captures and recognizes student actions in the classroom environment. To create our training, validation, and test datasets, we assign 80% of the video clips (amounting to 1,416 video clips) as the training dataset. Additionally, 5% of the clips (97 video clips) are used as the validation dataset to determine optimal hyperparameters. Finally, the remaining 15% of the clips (292 video clips) are reserved as the test dataset, enabling us to evaluate the performance of the final model accurately. We use AdamW optimizer [35] to optimize the student action recognition model with a learning rate of 1×10^{-3} during the linear probing stage and a learning rate of 1×10^{-6} during the finetuning stage. After the completion of training, we utilized the final trained student localization and student action recognition models, as outlined in Section III-C of our proposed student behavior understanding framework. These models were employed to perform student behavior understanding in our recorded 360-degree videos of classroom scenes.

B. Test 1: Student Localization

To begin, we present the numerical test results for our student localization model. Table II showcases the performance of our student localization model in comparison to two baseline models: RetinaNet [13] with ResNet-50 [26] as the backbone and SSD [12] with VGG-16 [28] as the backbone. Both baseline models followed the same training strategy as our student localization model, beginning with pretraining on the original COCO dataset [31] and then training on the student localization dataset while keeping all layers in the backbone model fixed. We conduct a comprehensive analysis by comparing various factors, including mean average precision (mAP), inference rate (samples/s) on the test set of the student localization dataset, floating point operations (FLOPs), and the number of trainable parameters (Param #).

Our observations reveal that our student localization model outperforms the baseline models in terms of mAP. Specifically, our model achieves an mAP that is 8.66% higher than that of RetinaNet [13] and 18.57% higher than that of SSD [12]. This signifies the superior accuracy of our student localization model in accurately identifying and localizing students within the classroom environment.

Moreover, our student localization model exhibits a reasonable computational complexity. When performing inference on the test dataset, it shows a 25.3% faster inference rate compared to RetinaNet [13], while being 52.3% slower than SSD [12]. In terms of floating point operations (FLOPs), our model demonstrates 14.7% fewer FLOPs than RetinaNet,



(a) (b) (c)

Fig. 6: Example results of our student behavior analysis workflow: (a) correct analysis of the action of a student, (b) wrong analysis of the action of a student, and (c) a student back to the camera.

Model	mAP (%)	Inference rate (samples/s)	GFLOPs	Param #
Ours	**43.28**	18.3	197.2	41.8M
RetinaNet	34.62	14.6	231.3	34.1M
SSD	24.71	38.4	34.9	35.6M

TABLE II: Comparison of different object detection models on the student localization dataset.

indicating more efficient processing, while still utilizing 5.6× more FLOPs than SSD [12].

Overall, the test results affirm the superiority of our student localization model, surpassing the baseline models in terms of mAP while maintaining a reasonable computational complexity.

C. Test 2: Student Action Recognition

We now proceed to present the results of our student action recognition experiment. Table III showcases the performance of our model in comparison to five baseline models: three vision-based action recognition models (X3D-L [36], SlowFast [37] with ResNet50 [26] as the backbone, and Video Vision Transformer (ViViT) [38]) and two skeleton-based action recognition models (CTR-GCN [39] and MS-G3D [40]).

For the vision-based action recognition models, all baseline models were pretrained on the Kinetics-400 dataset and subsequently finetuned on our student action dataset. The skeleton-based action recognition models, on the other hand, were pretrained on the NTU-RGB+D-120 dataset [41] and finetuned using the skeleton data extracted from our student action dataset using AlphaPose [42].

We conducted a comprehensive comparison, focusing on performance metrics, especially accuracy on the test set of the student action dataset, as well as computational complexity indicators, including inference rate (frames/s) on the test set, floating point operations (FLOPs) during inference, and the number of trainable parameters (Param #).

In our analysis, we initially compared our model with the three vision-based action recognition models. Our model demonstrated the best performance, achieving an accuracy of 96.14%, while maintaining a reasonable computational cost, with an inference rate of 232.3 frames/s and 170.4 GFLOPs. The closest performing baseline model was ViViT, with an accuracy only 4.62% lower than our model. However, ViViT [38] came with a significantly higher computational complexity, with 1.4× more FLOPs than our model, and a much slower inference rate, being 3.3× slower than ours. X3D-L [36] exhibited a performance of 16.01% degradation than our model, but with a faster inference rate (41.3% faster than ours) and lower computational complexity (3.3 × fewer FLOPs than our model). SlowFast [37] with ResNet50 [26] as the backbone achieved a performance of 22.32% degradation than our model, with lower computational cost (2.35× fewer FLOPs) and a faster inference rate (41.3% faster than our model).

Next, we compared our model with the two skeleton-based action recognition baseline models. We observed that skeleton-based models had a significantly smaller number of trainable parameters, with CTR-GCN having 1.4M parameters and MS-G3D having 3.2M parameters, compared to our model with 36.6M parameters. Additionally, skeleton-based models demonstrated faster inference rates, with CTR-GCN being 3.06× faster and MS-G3D being 2.61× faster than our model. However, the performance of skeleton-based models was notably worse than our model, with CTR-GCN being 43.67% worse and MS-G3D being 38.96% worse. This indicates that skeleton-based models are unsuitable for our dataset.

In conclusion, our model outperformed all baseline models. It achieved superior performance with a reasonable computational cost, demonstrating its suitability for student action recognition tasks in our dataset.

Model	Acc (%)	Inference rate (frames/s)	GFLOPs	Param #
Ours	**96.14**	232.3	170.4	36.6M
X3D-L	80.13	328.3	39.5	5.3M
SlowFast+R50	73.82	353.9	50.9	33.6M
ViViT	91.52	54.2	413.7	133.1M
CTR-GCN	52.47	943.7	8.7	1.4M
MS-G3D	57.18	837.6	16.9	3.2M

TABLE III: Comparison between our action recognition model and other baseline action recognition models.

Action	TP #	FP #	Precision
Manipulate and count	41	11	0.79
Write	103	21	0.83

TABLE IV: Numerical result of our student behavior analysis workflow on the 360-degree classroom environment videos.

D. Test 3: Student Behavior Understanding

Our final experiment involves testing our student behavior understanding framework on our recorded real-world 360-degree classroom environment videos. To understand student behavior in these videos, we followed our proposed framework in Section III-C. After conducting our analysis, we manually checked the results and counted the number of true positive (TP) and false positive (FP) recognized actions. We then calculate the precision of each action using the formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{8}$$

The results are shown in Table. IV. We observed that our workflow was able to accurately capture students and their behaviors in most cases. An example of an accurately analyzed result is shown in Fig. 6 (a). However, when students performed actions that were not included in our dataset, such as talking, our model would randomly classify the action as either writing, manipulating, or counting, as shown in Fig. 6 (b). Additionally, when students had their backs to the camera, our model would always misclassify their actions, as shown in Fig. 6 (c). This is likely because our training videos only included actions performed in front of the students, and did not capture actions performed with the student's back facing the camera.

V. Conclusion and Future Work

Conclusion. In this work, we present a deep learning-based framework that effectively leverages object detection and action recognition techniques to gain a comprehensive understanding of student behavior in classroom environments. To facilitate this research, we also introduce a novel video dataset specifically featuring students' actions within the classroom environment. We commence our evaluation by assessing the performance of the two individual components: the object detection model and the action recognition model, using our proposed datasets. Subsequently, we conduct a comprehensive evaluation of the framework as a whole, utilizing a collection of recorded 360-degree classroom environment videos. Through meticulous comparison and analysis, we demonstrate that our proposed framework excels in accurately capturing students and recognizing their actions. The results highlight the potential of our framework to significantly enhance teaching practices in classrooms. By leveraging the power of deep learning techniques and employing innovative methodologies, our research provides valuable insights into student behavior analysis and presents a promising avenue for further improving educational settings.

Future work. (1) One of the weaknesses in our work lies in the inability of our student localization model to differentiate between students and teachers. This limitation arises due to the training dataset, which only distinguishes between persons and non-person objects. To address this, enhancing the current student localization dataset by incorporating specific categories for students and teachers and augmenting the dataset with a larger collection of student images would greatly improve the performance of our framework. (2) Another area for further improvement is enhancing the generalization ability of our framework. Currently, our student action recognition dataset predominantly focuses on capturing students from the front view. However, in real classroom scenarios, students can adopt various angles and poses in relation to the camera. To address this limitation, collecting a diverse set of unlabeled videos featuring students in different angles and engaging in more complex actions would aid in enhancing the generalization ability of our framework. Additionally, incorporating reinforcement learning with human feedback (RLHF) techniques could further refine our framework's ability to accurately recognize student actions. (3) Exploring additional features, especially the analysis of student facial expressions and recorded audio in the classroom environment, can provide a more comprehensive understanding of student behavior in classroom scenes. Investigating these aspects will be a focus of our future work. (4) Last but not least, applying our proposed framework to real-world applications, particularly integrating web-based virtual reality techniques, where educators can freely access recorded or streaming 360-degree classroom environment videos with students and their actions annotated using our framework, presents another major direction for future work. By addressing these aspects in future work, we can further strengthen our framework's ability to differentiate between students and teachers, improve its generalization capabilities, and ultimately enhance its performance in real-world classroom environments.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant #2217104, #2212465, and DRK-12 #1908159. The IRB number used for this study is #19-232. NIST's role was limited to activities not involved with the human subjects research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Institute of Standards and Technology. Certain commercial products are identified in this paper in order to specify the experimental procedure adequately. Such

identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

References

[1] P. Grossman, C. Compton, D. Igra, M. Ronfeldt, E. Shahan, and P. W. Williamson, "Teaching practice: A cross-professional perspective," *Teachers college record*, vol. 111, no. 9, pp. 2055–2100, 2009.

[2] T. B. Abdallah, I. Elleuch, and R. Guermazi, "Student behavior recognition in classroom using deep transfer learning with vgg-16," *Procedia Computer Science*, vol. 192, pp. 951–960, 2021.

[3] K. Altuwairqi, S. K. Jarraya, A. Allinjawi, and M. Hammami, "Student behavior analysis to measure engagement levels in online learning environments," *Signal, Image and Video Processing*, vol. 15, no. 7, pp. 1387–1395, 2021.

[4] B. Yang, Z. Yao, H. Lu, Y. Zhou, and J. Xu, "In-classroom learning analytics based on student behavior, topic and teaching characteristic mining," *Pattern Recognition Letters*, vol. 129, pp. 224–231, 2020.

[5] F.-C. Lin, H.-H. Ngo, C.-R. Dow, K.-H. Lam, and H. L. Le, "Student behavior recognition system for the classroom environment based on skeleton pose estimation and person detection," *Sensors*, vol. 21, no. 16, p. 5314, 2021.

[6] M. A. Rafique, F. Khaskheli, M. T. Hassan, S. Naseer, and M. Jeon, "Employing automatic content recognition for teaching methodology analysis in classroom videos," *Plos one*, vol. 17, no. 2, p. e0263448, 2022.

[7] Y. Huang and M. Liang, "Spatio-temporal attention network for student action recognition in classroom teaching videos," 2021.

[8] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[9] T. Nguyen, E. Park, J. Han, D.-C. Park, and S.-Y. Min, "Object detection using scale invariant feature transform," in *Genetic and evolutionary computing*. Springer, 2014, pp. 65–72.

[10] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," *Advances in neural information processing systems*, vol. 26, 2013.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[14] S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," *Advances in neural information processing systems*, vol. 22, 2009.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[16] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[21] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6824–6835.

[22] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3202–3211.

[23] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[24] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[25] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[32] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

[33] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[34] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=UYneFzXSJWh

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[36] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 203–213.

[37] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6202–6211.

[38] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.

[39] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 359–13 368.

[40] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 143–152.

[41] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

[42] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.