VMAPN: A Vision Mamba-Based Real-Time Behavior Recognition and Feedback System for Smart Education

Zili Chen[1,3], Aihong Liu[2*]
[1]School of Artificial Intelligence and Big Data, Chongqing College of Science and Creation, Yongchuan, 402160, China
[2]Fan Changjiang School of Journalism, Neijiang Normal University, Neijiang, 641100, China
[3]College of Physics and Information Engineering, Zhaotong University, Zhaotong, 657000, China.
E-mail: AiHongLiuu@outlook.com
[*]Corresponding author

Keywords: vision mamba, intelligent education, real-time feedback, learning behavior recognition, teaching evaluation Model (TEM), Augmented Prototypical Network (APN)

Received: April 29, 2025

With the growing demand for intelligent educational technologies, traditional methods of classroom evaluation—often reliant on subjective observations—fall short in scalability, objectivity, and timeliness. This paper proposes a real-time automatic scoring and feedback system for intelligent educational platforms, powered by the Vision Mamba architecture. Vision Mamba—a novel and lightweight spatio-temporal modeling backbone—is integrated with an enhanced classification module, the Augmented Prototypical Network (APN) to form the VMAPN framework, which enables accurate behavior recognition under small-sample conditions. The system analyzes classroom video streams to identify student behaviors such as attentiveness, participation, and posture, and generates adaptive feedback for both students and teachers. Real-time feedback is delivered through an interactive interface, while backend analytics leverage machine learning techniques to monitor learning engagement and evaluate teaching effectiveness. Furthermore, the proposed Teaching Evaluation Model (TEM) classifies student behaviors into four categories—positive, relatively positive, neutral, and negative—to derive objective teaching effect scores. Experimental results on THUMOS 2014 and ActivityNet v1.3 datasets validate the model's predictive performance, achieving mAP scores of 64.7% at IoU 0.5 on THUMOS 2014 and 72.3% at IoU 0.5 on ActivityNet v1.3, representing improvements of 6.2% and 4.8%, respectively, over baseline methods.

Povzetek: Predlagan je sistem, ki v realnem času iz video posnetkov samodejno prepoznava vedenje učencev ter učiteljem in učencem zagotavlja objektivno povratno informacijo in oceno pouka.

1 Introduction

In traditional classroom teaching, evaluation often relies on expert observations and manually filled evaluation forms, which are typically summarized over multiple sessions to assess instructional effectiveness. Although such expert-based evaluations play a supervisory role, they are inherently limited by subjectivity, randomness, and inefficiency. With the rapid development of artificial intelligence (AI), novel approaches have emerged that allow for real-time, objective, and scalable assessment of classroom teaching by analyzing student behavior and engagement through data-driven models [1].

Recent advances in computer vision and deep learning have enabled automated recognition of learning behaviors from video data [2]. These technologies can be applied to intelligently monitor classroom interactions, analyze student attentiveness, engagement, and participation, and ultimately provide meaningful insights into teaching quality. Compared with traditional methods such as time-sampling, manual coding (e.g., FIAS, ITIAS) [3], or ethnographic observation, AI-powered behavior analysis systems offer greater consistency, scalability, and automation, significantly reducing the burden on educators and improving the timeliness of feedback. Despite the increasing integration of online education, the physical classroom remains the primary environment for large-scale youth education [4]. However, the digitization and informatization of classroom assessment lag behind due to reliance on inefficient manual methods. Meanwhile, international and national policy initiatives—such as China's Education Informatization 2.0 Action Plan and UNESCO's AI in Education—emphasize the importance of smart education platforms, encouraging the adoption of AI, deep learning, and intelligent feedback systems in education.

In response to these challenges and policy directions, this study proposes a real-time automated assessment and feedback system for intelligent education platforms, powered by the Vision Mamba architecture. Vision Mamba, as a cutting-edge vision backbone model,

provides efficient and robust spatio-temporal modeling capabilities suitable for video-based behavior recognition. Leveraging this architecture, we design a novel framework that integrates deep video behavior recognition with AI-based decision-making, enabling real-time evaluation of classroom learning activities and personalized feedback delivery. Additionally, we address the growing demands of online assessments, where large-scale exam proctoring suffers from inadequate human supervision. By incorporating intelligent video analysis, the proposed system can automatically detect cheating behaviors during online examinations, alert invigilators in real-time, and improve the fairness and reliability of online assessments. This not only alleviates the monitoring burden on educators but also significantly enhances the scalability and integrity of online education systems.

In summary, this paper investigates a Vision Mamba-powered intelligent education platform capable of automatically scoring and providing feedback in real-time. By integrating advanced video behavior recognition techniques with scalable AI systems, we aim to bridge the gap between traditional and smart classroom evaluation, promote educational informatization, and support precise, data-driven instruction and assessment strategies. To address these challenges, the main objectives of this study are as follows:

(i) Validate the effectiveness of Vision Mamba for behavior recognition tasks under small-sample conditions in educational settings;

(ii) Construct a scalable, real-time scoring and feedback system integrating behavior recognition with interactive interfaces;

(iii) Benchmark the proposed VMAPN framework against classical backbone models (e.g., ConvNets, ViT) on educational behavior datasets such as THUMOS 2014 and ActivityNet v1.3.

2 Related work

2.1 Video-based behavior recognition

The task of behavior recognition from video data is significantly more complex than image-based analysis due to the temporal dimension inherent in video. Three mainstream approaches dominate current research in video understanding: two-stream networks [5], 3D convolutional networks [6], and self-attention-based models [7]. Two-stream networks laid the foundation for video classification by processing spatial and temporal features separately spatial features via RGB frames and motion via optical flow [8]. Follow-up works introduced various fusion strategies and segment-based temporal modeling to enhance long-term temporal understanding. Despite their success, two-stream methods suffer from high computational and storage costs due to the pre-computation of optical flow, limiting their real-time applicability. 3D convolutional networks address video modeling by extending 2D convolutions across the temporal axis [9]. Models like C3D, I3D, and R(2+1)D have demonstrated promising performance on standard video benchmarks [10]. Efficient variants such as X3D and MoViNet optimize latency and memory usage, enabling real-time inference [11]. Recent innovations like SlowFast and TDN further enhance temporal resolution by introducing multi-scale or dual-pathway designs [12]. Self-attention-based models, especially those adapted from the Transformer architecture, offer superior global temporal reasoning. Non-local networks, TimeSformer, and MViT utilize token-based representations to model long-range dependencies [13]. Although these approaches outperform previous architectures in accuracy, their high computational demands necessitate large-scale datasets and careful optimization. Lightweight alternatives such as X-ViT and MorphMLP seek to balance efficiency with performance through architectural simplifications or prompt-based learning [14].

2.2 Smart education and student behavior analysis

As AI-powered education platforms become increasingly prevalent, researchers are exploring how computer vision techniques can be used to quantify and analyze student behaviors in real classroom settings. Traditional manual coding and observation methods, such as FIAS or ethnographic analysis [15], are insufficient for large-scale implementation due to subjectivity and labor intensity. To address this, several interdisciplinary efforts have emerged. Researchers have utilized Kinect-based skeleton tracking combined with machine learning classifiers for classroom posture recognition [16]. Other works adopted multimodal approaches—integrating speech, body movement, and head tracking—to improve interaction analysis and engagement detection. Advances in deep learning have further enabled end-to-end student behavior recognition systems based on CNNs, residual networks, and GANs, achieving high classification accuracy and fast convergence. In online and hybrid classroom contexts, gesture and pose recognition have been successfully employed to track participation and detect behaviors such as hand-raising, standing, or sleeping. For instance, OpenPose-based models have demonstrated robust skeletal tracking for real-time activity classification [17]. Recent studies have explored fine-grained behavioral datasets for supervised learning tasks using deep residual networks, CNN-10, and ensemble models such as XGBoost, reaching behavior classification accuracies upwards of 93% [18]. Recent AI-driven feedback systems have explored multi-modal data integration, offering promising results for adaptive learning [26, 27].

2.3 Pose estimation and classroom activity recognition

Pose estimation has become a key enabler of student behavior recognition. Enhanced OpenPose-based approaches and multi-view 3D pose estimation systems improve accuracy and reliability [19], even in occluded or crowded classroom environments. Methods combining

pose estimation with deep learning models [20], such as CNNs or Boosting frameworks, have shown excellent performance in real-world scenarios, with several models achieving over 90% accuracy on customized classroom datasets. The integration of pose-based analysis with spatio-temporal models provides a promising direction for building scalable [21], real-time educational assessment systems. However, many existing methods still face limitations in generalization across diverse classroom layouts, lighting conditions, and student postures. These challenges motivate the need for lightweight, real-time, and generalizable architecture—such as Vision Mamba, capable of efficient behavioral modeling and feedback in intelligent education systems.

To provide a concise comparison of mainstream video-based behavior recognition methods, Table 1 summarizes key architectural approaches, tasks, datasets, and performance metrics, highlighting the advantages of Vision Mamba for small-sample educational scenarios.

Table 1: Comparative summary of video-based behavior recognition approaches

Method	Architecture Type	Target Task	Datasets Used	mAP / Acc
C3D [9]	3D CNN	Action Classification	UCF101, Sports-1M	~51.6%
I3D [10]	3D CNN (Inflated)	Action Recognition	Kinetics-400, UCF101	~66.4%
SlowFast [12]	Dual-pathway CNN	Action Recognition	Kinetics-400, AVA	~77.9%
TimeSformer [13]	Transformer-based	Temporal Reasoning	Something-Something V2	~62.1%
VMamba [22]	State-Space Model	Spatio-temporal Modeling	Synthetic, ImageNet-1K	~73.4%
Vision Mamba (ViM)	Bi-directional SSM	Behavior Recognition	THUMOS 2014, ActivityNet	64.7% / 72.3%

3 Method

To achieve real-time automated scoring and feedback in intelligent education platforms, we propose an advanced framework leveraging the Vision Mamba architecture. This section details the methodology underlying our proposed system, including feature extraction, model architecture, classification process, loss computation, and pretraining strategies.

3.1 Architecture overview

The overall framework of the Vision Mamba-based Automated Scoring and Feedback System (VMASFS) is illustrated in Figure 1. The system consists of two primary components: feature extraction using Vision Mamba and automated decision-making through an Augmented Prototypical Network (APN). Vision Mamba (ViM) serves as the backbone model, extracting spatio-temporal features from video sequences of classroom activities and student behaviors. These extracted features are then fed into the Augmented Prototypical Network (APN) for behavior classification, engagement scoring, and real-time feedback generation.

The proposed framework effectively integrates global and local feature representations. The global category-labeled features capture high-level semantic information about the overall student engagement, while local block features provide fine-grained analysis of individual postures, facial expressions, and interactions. By fusing these features, the system enhances behavior recognition accuracy and instructional feedback precision.

The final step in the framework involves real-time decision adjustment through an adaptive scoring mechanism, ensuring the generated feedback is both personalized and pedagogically relevant. This intelligent assessment and feedback loop facilitates a scalable and automated evaluation process, reducing the reliance on manual observation while improving the consistency and objectivity of educational assessments.

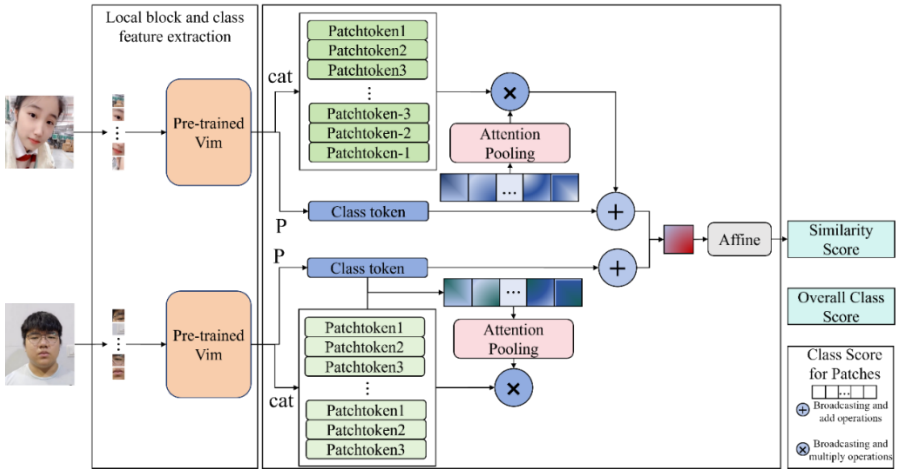

Figure 1: Overview of the VMAPN-based real-time behavior classification and feedback model.

3.2 VMAPN network design

The network architecture of VMAPN consists of a feature embedding module and a classification module. The feature embedding module is dedicated to creating an embedding space that can transform raw data into information-rich feature vectors. Traditional small-sample classification algorithms usually use smaller network architectures, such as Conv-4 and ResNet-12, to ensure fairness in comparison. However, with the rapid development of computer vision technology, these architectures have become obsolete and may lead to poor classification accuracy. To solve this problem, subsequent studies have introduced larger and more advanced network architectures, such as ViT, which provide significant improvements in small-sample classification accuracy. Current ViT-based methods are nearing saturation, and a large number of studies have begun to explore further optimization of classification modules based on ViT and its variants.

This study explores a new network architecture used as a feature embedding module for small samples. Recently, an improved state-space model, Mamba [22], has received a lot of attention. Mamba improves on the S4 model by fusing time-varying parameters, which enables the model to dynamically select relevant information based on the input data. In addition, Mamba introduces a hardware-aware algorithm to improve the execution efficiency of the algorithm. The Mamba model performs strongly in natural language processing tasks and has recently been adapted for visual applications. For example, Vision Mamba (ViM) integrates Mamba into a Transformer-like architecture to address visual challenges using a bi-directional state-space model. In contrast, VMamba introduces a cross-scanning mechanism that links one-dimensional sequences with two-dimensional image structures.

Although there is no study that directly applies Mamba to small-sample scenarios, the above discussion demonstrates the great potential of Mamba for small-sample learning. the time-varying parameters of Mamba allow the model to dynamically adjust its internal parameters according to the input data. This flexibility is particularly suitable for dealing with small-sample data because it can effectively utilize the limited data resources to filter out irrelevant information, while adjusting its behavior according to the needs of each specific task to optimize the whole learning process, which will enhance the model's ability to generalize across different tasks. Moreover, experiments have proved that Mamba has strong modeling capability. Compared to ViT, Mamba achieves a new balance of speed and accuracy, with lower computational complexity while maintaining efficient performance. The main goal of this study is to explore and validate the application of Mamba in small-sample learning, and further optimization of speed will follow.

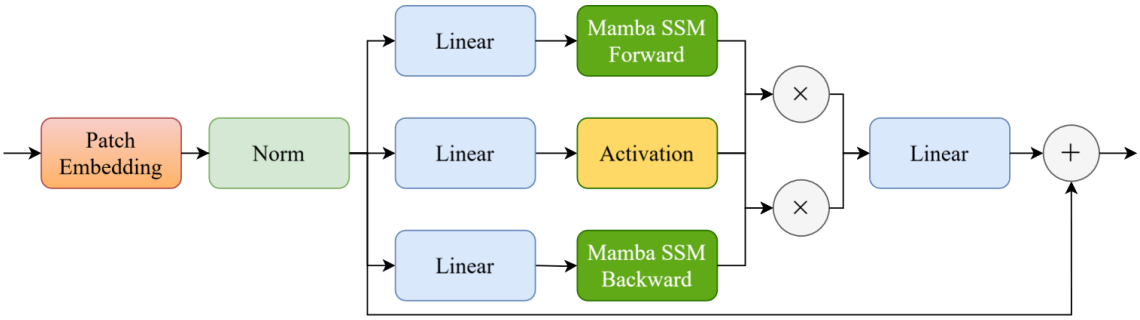

Figure 2: Bi-SSM architecture diagram

In this paper, ViM is adopted as the feature embedding module to address the small-sample classification challenge. Although ViM is not a Transformer, its architecture borrows certain structural elements—such as patch-based input encoding and positional embeddings—that allow for a fair comparative baseline with ViT. Unlike ViT, which relies on self-attention to capture long-range dependencies, ViM uses a bidirectional state-space model (Bi-SSM) to achieve global contextual awareness. This design enables ViM to process sequences with lower computational complexity while still capturing temporal and spatial relationships. Importantly, while ViM shares some high-level architectural traits with Transformers, it does not employ attention mechanisms, nor does it introduce image-specific inductive biases. Instead, its performance stems from dynamic sequence modeling and data-dependent parameterization. Further, research [23] has provided supervised pre-training of ViM on ImageNet1K. The applicability of ViM is not limited to supervised learning tasks, and this study extends ViM to self-supervised tasks by pre-training ViM with masked image modeling to obtain more general features. This pre-training strategy can further improve the accuracy of the model in small-sample classification, allowing it to learn from limited data and generalize to new and unseen tasks more efficiently.

The feature vectors extracted by the feature embedding module are fed into the classification module for classification. The prototype network is the baseline method for the small-sample classification module, which categorizes query samples by comparing the feature similarity between the query samples and the category prototypes. The category prototype is obtained by averaging the features of all samples in the same category. These sample features can be represented as single vectors obtained from the feature maps compressed by the network, or as category labeled features generated by a ViT-like architecture. The category labeled features aggregate the global information of the entire image, and thus are often used as the basis for the final classification decision. In small sample scenarios, due to the limited number of samples, the global information of each category is much less rich than that in large-scale datasets. Therefore, we need to make full use of the available information, such as the local block features of each local region after image segmentation. These features cannot be fully obtained by relying only on the category tagging features under the small sample limitation. Fusing the local block features into the category labeling features can provide a richer selection of features for small-sample classification. It is worth noting that not all local block features are useful for the classification task, such as large background areas or areas not related to category labels, which may be irrelevant local image blocks and need to be filtered out to find out the class-related local block features.

Based on this, this study improves the prototype network to further enhance the performance of small-sample image classification by connecting the Augmented Prototypical Network (APN) to the output of Vi M. By combining the efficient features of Vi M with the APN, the APN can be used for the classification of small-sample images, and the APN can be used for the classification of small-sample images. The design of VMAPN is based on Vision Mamba and Augmented Prototypical Network by combining the efficient feature extraction capability of Vi M with the fine classification capability of APN, which is designed to take into account the representativeness of the category labeling features and the importance of the local block features in the small-sample scenarios. Combining local block features to refine the category labeling vector is a better choice. By integrating features from different levels or regions into the category labeling features, the information of the image can be expressed more comprehensively and a richer prototype representation can be constructed. After each input image $x \in R^{H\times W\times 3}$ is processed by Vision Mamba (ViM), the model outputs; A global category-labeled feature vector $\mathbf{f}_g \in \mathbb{R}^d$, where $d = 768$ is the embedding dimension; A set of local block features $\{f_j\}_{j=1}^{K}$, with each $f_j \in R^d$, extracted from *K* spatial positions in the feature map (e.g., $K = 49$ for a 7×7 grid).

The similarity between each local feature f_j and the global feature f_g is computed using cosine similarity:

$$s_j = \frac{f_j \cdot f_g}{|f_j| \cdot |f_g|}, j = 1,2, \dots, K$$

The weights α_j for each local block are obtained by applying the softmax function over all similarities:

$$\alpha_j = \frac{\exp(s_j)}{\sum_{k=1}^{K} \exp(s_k)}$$

The weighted local feature representation is then computed as:

\[

$$f_{local} = \sum_{j=1}^{K} \alpha_j f_j$$

\]

The final feature vector $\mathbf{f} \in \mathbb{R}^{2d}$ used for classification is obtained by concatenating the global feature and the weighted local feature:

$$f = f_g \oplus \boldsymbol{f}_{local} \in R^{2d}$$

In the support set, class prototypes are computed by averaging the feature vectors of all support samples in each class. During inference, each query feature vector is compared to all class prototypes using a similarity metric (e.g., Euclidean or cosine distance), and a classification score is computed accordingly. An affine transformation is applied to adjust the decision boundary:

$$S = scale * \left[metric\left(f_{class}, f_{query}\right) + bias\right]$$

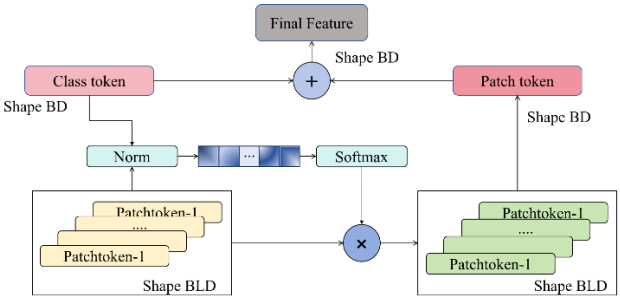

Figure 3: APN architecture diagram

3.3 Loss calculation

The similarity score matrix S is converted to a log-probability matrix P. The similarity score s_{ij} for the ith query sample in S and the jth support set category is converted to a log-probability $\log(p_{ij})$ in the following manner, where N is the number of categories.

$$log(p_{ij}) = log\left(\frac{exp(s_{ij})}{\sum_{j=1}^{N} e\,xp(s_{ij})}\right)$$

Design a label-smoothed one-hot matrix Y with true category labels $y_i^Q \in \{1,2,\ldots,N\}$ for the i th query sample, with one-hot taking the value y_{ij} as:

$$y_{ij} = \begin{cases} 1-\varepsilon & if(j = y_i) \\ \dfrac{\varepsilon}{N-1} & otherwise \end{cases}$$

where ε is a smoothing parameter that takes the value of 0.1. The cross-entropy loss is computed using the label-smoothed one-hot matrix Y with the log-probability matrix P.

$$L = -\frac{1}{Q}\sum_{i=1}^{q}\sum_{j=1}^{N} y_{ij} * log(p_{ij})$$

3.4 Pre-training strategies

Pre-training techniques have become a key strategy in the field of small-sample learning. The generalization ability and learning efficiency of the model can be effectively improved by appropriate pre-training techniques. Literature [22] has achieved significant results in self-supervised pre-training of Vision Transformer on MiniImageNet. Literature [10] explores the impact of different combinations of network structures and pre-training techniques on small sample learning on larger datasets such as ImageNet-1K. Supervised pre-training shows significant advantages with large-scale datasets, where a large amount of data provides diverse samples that can help ViM learn more generalized feature representations, thus showing good results on unknown data as well. However, it is less effective on small-scale datasets, where the restricted labeling information tends to lead to overfitting, making the model usually less adaptable and generalizable to new categories.

In this study, we adopted two pretraining strategies. For the supervised setup, we utilized publicly available pretrained weights of ViM on ImageNet-1K, which includes approximately 1.2 million labeled images and was trained for 300 epochs. For the self-supervised approach, we followed the method used in literature [17] and performed masked image modeling (MIM) on the MiniImageNet training set (~100,000 images), using 40% random masking over 200 training epochs. These settings allow the model to capture more generalizable features when labels are scarce.

By dividing the image into multiple patches, randomly masking certain patches, and letting ViM encode and reconstruct these patches, the model can develop a deeper understanding of image structure and content. This helps it learn generic underlying features, rather than merely class-specific signals. This approach provides a more adaptive learning mechanism for small-sample classification tasks and is especially effective when labeling data is limited. In this study, both supervised and self-supervised pretraining strategies were explored to assess their suitability for small-sample scenarios. However, for the VMAPN model evaluated in Section 5, we exclusively used the publicly available supervised ViM weights pretrained on ImageNet1K, due to resource and reproducibility constraints. The discussion of self-supervised masked image modeling (MIM) on MiniImageNet is presented as a potential direction and proof of concept, not part of the current system's deployed configuration. Future work will include comparative training using self-supervised ViM variants to validate performance gains in low-label environments.

4 Design of automatic grading and feedback system for educational platforms

To implement the proposed Vision Mamba-powered intelligent behavior recognition framework in a real-world educational context, this section introduces the comprehensive design of an automatic scoring and feedback system suitable for both classroom instruction and online assessments. The system integrates frontend interface design, backend data processing, real-time behavior recognition, and intelligent feedback generation. It is built on a scalable and modular architecture that supports both usability and computational efficiency. Furthermore, the system takes into account the distinct roles and expectations of students and teachers, ensuring personalized feedback and decision support across different teaching scenarios.

4.1 User role definitions and functional requirements

The system is designed with two primary user roles in mind: student users and teacher users. The needs of each group were systematically analyzed to guide the system's functional and interface design. For student users, ease of access and clarity of interaction are essential. Students require a simple yet informative interface where they can easily participate in assessment tasks and receive detailed feedback on their performance. The feedback must be timely, personalized, and grounded in objective behavioral analysis, helping students identify their strengths and weaknesses in a data-informed manner.

Teacher users, on the other hand, require broader control over assessment tasks, student data, and feedback

customization. Their core expectations include the ability to publish and modify tests, monitor student engagement in real-time, and analyze behavior trends across sessions. Furthermore, teachers expect the system to assist in grading, detect behavioral anomalies, and generate high-level instructional recommendations. To fulfill these expectations, the system integrates intelligent data pipelines that transform video-based behavior recognition into practical, actionable educational insights.

4.2 System architecture and technical stack

The overall system architecture consists of three tightly integrated layers: the user-facing frontend, the intelligent behavior recognition engine, and the backend data management and analysis module. The entire platform is developed using Python, with Django serving as the central web framework due to its rapid development capabilities, integrated ORM, and support for user authentication and routing.

The frontend interface employs standard web development technologies including HTML, CSS, JavaScript, and Bootstrap, ensuring responsive design and cross-platform compatibility. Asynchronous JavaScript and XML (AJAX) is used to enhance user interaction by allowing page updates without full reloads, which is particularly important for displaying real-time feedback and notifications during assessments.

At the core of the system lies the Vision Mamba and Augmented Prototypical Network (VMAPN) module, responsible for extracting spatio-temporal features from classroom or examination video streams. These features, including attention level, posture stability, and gesture activity, are processed in real-time and fed into the scoring engine. The backend supports secure data storage, task processing, and intelligent analytics using a combination of Pandas for data transformation and Scikit-learn for machine learning-based prediction and clustering. This structure allows the system to provide real-time analysis while maintaining long-term performance tracking and personalization. The general architecture of the system is illustrated in Figure 4, highlighting the interaction between user interfaces, backend modules, data processing pipelines, and the Vision Mamba-powered intelligent behavior engine.

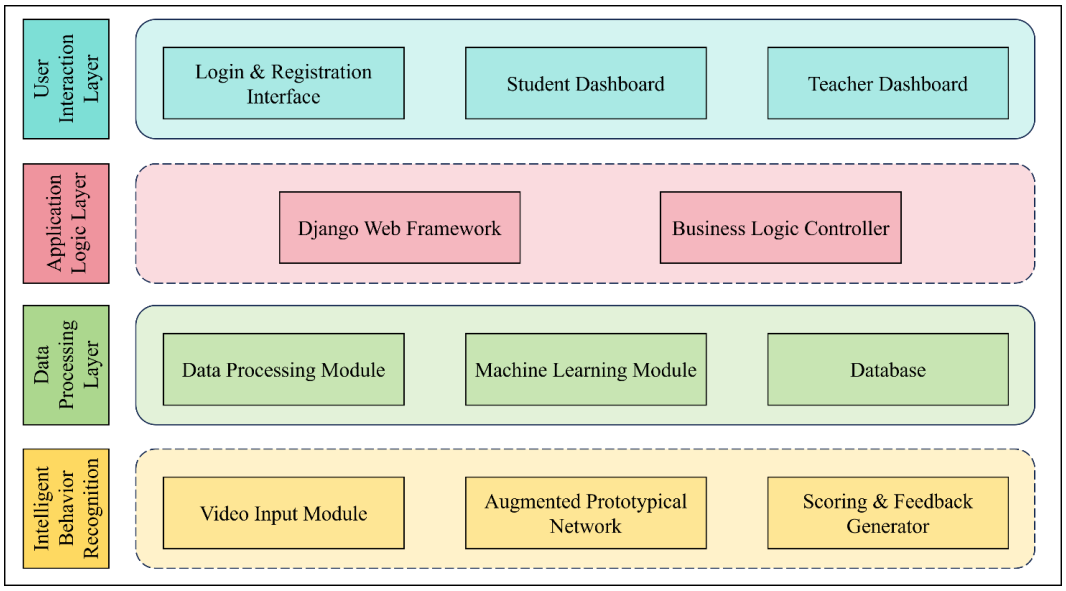

Figure 4: General architecture of the real-time automatic scoring and feedback system based on Vision Mamba and APN.

4.3 Frontend design for intelligent interaction

The frontend of the system is structured into three main interfaces: a unified login and registration portal, a student interaction dashboard, and a teacher management console. The login and registration module supports role-based access control and uses secure authentication mechanisms to manage users and protect data. Once authenticated, users are redirected to role-specific environments.

The student dashboard is designed to facilitate seamless participation in evaluations, access real-time feedback, and track personal progress. Upon completing an assessment task, the system immediately displays behavioral scores generated by the VMAPN model, along with visual feedback such as attention timelines, engagement heatmaps, and suggestions for improvement. This feedback enables students to reflect on their learning habits and adjust accordingly.

The teacher dashboard provides task management capabilities including test creation, question editing, scheduling, and evaluation settings. Teachers can also view student behavior analytics in aggregated or individual form. Real-time alerts for inattention or anomalous behavior (e.g., potential cheating during online assessments) are displayed prominently to support immediate pedagogical intervention. The interface also

enables teachers to download session reports and receive system-generated suggestions for improving classroom interaction based on behavioral trends.

4.4 Backend system design and intelligent analytics

The backend is the core operational layer of the system and is responsible for data processing, behavior evaluation, and task logic management. It comprises three essential modules: the database layer, the data processing pipeline, and the business logic controller.

The database design follows a relational model and supports core entities such as User, Course, Task, VideoLog, Score, and Feedback. Data normalization, indexing, and foreign key constraints are applied to ensure consistency and retrieval efficiency. Data integrity is enforced through strict validation rules, and backups are automated to guarantee recovery under system failure.

The data processing module uses Pandas to preprocess data collected from behavior analysis, such as converting timestamps, handling missing values, and encoding categorical variables. The Scikit-learn library supports multiple analytics workflows, including regression-based performance prediction and K-means clustering for engagement profiling. For example, a regression model may predict test scores based on attendance, attention scores, and prior performance. A clustering algorithm may segment students into high, medium, and low engagement groups, helping teachers apply differentiated instructional strategies.

Most importantly, the behavior recognition engine leverages the Vision Mamba backbone for efficient spatio-temporal modeling. Video frames are processed in real time to extract both global category-labeled features and local block features. These are fed into the Augmented Prototypical Network, which performs similarity-based classification using weighted aggregation of relevant local blocks. The final feature representation is compared with class prototypes, and classification scores are adjusted through an affine transformation before being passed to the feedback module.

To ensure real-time performance, the backend system adopts multithreaded task queues using Python's concurrent.futures.ThreadPoolExecutor for parallel processing of video frames, behavior inference, and database writes. Caching mechanisms are implemented via Redis for frequently accessed inference outputs and session metadata, reducing repeated computation and database load. The behavior recognition module is accelerated using GPU inference (NVIDIA RTX 3060, 12GB VRAM) via PyTorch’s CUDA backend. During deployment, the system achieved an average inference latency of 43.7 ms per frame, corresponding to ~22.9 FPS, which is sufficient for real-time classroom analysis. Average CPU utilization remained below 40%, and RAM usage peaked at 6.3 GB during concurrent student evaluations. These results demonstrate the system’s capability for smooth real-time deployment on mid-range hardware without introducing user interface lag or backend bottlenecks.

4.5 Real-time feedback generation and visualization

A core innovation of the system lies in its ability to transform raw behavioral data into meaningful, real-time feedback. For students, this includes the immediate display of scores across multiple behavioral dimensions such as attentiveness, participation, posture correctness, and activity. Feedback is presented through intuitive visualizations like attention span graphs, session heatmaps, and time-series engagement curves. These tools are especially helpful in promoting self-regulated learning and allowing students to reflect on their performance patterns.

For teachers, the system offers real-time analytics dashboards with aggregated engagement data, anomaly detection alerts, and performance distributions. During online assessments, the system can detect and flag suspicious behaviors such as screen avoidance, frequent turning, or multi-device usage, thus supporting real-time intervention and reducing the reliance on manual proctoring. After each session, teachers receive a system-generated report summarizing behavioral patterns, participation trends, and actionable insights tailored to instructional improvement.

This intelligent feedback mechanism is made possible by the lightweight yet expressive modeling capabilities of Vision Mamba architecture. Unlike traditional Transformer-based models, Vision Mamba achieves global contextual understanding through a bidirectional state-space model, making it particularly well-suited for real-time educational scenarios. Its integration with the APN module further enhances the feedback system by enabling finer behavioral categorization under limited data conditions, making it ideal for small classroom settings or targeted assessments.

5 Results and discussion

5.1 Experimental data collection

To support both general video action recognition and classroom-specific behavior analysis, two types of datasets were utilized. First, for benchmarking the VMAPN architecture on standard datasets, we used THUMOS 2014 and ActivityNet v1.3, both of which contain annotated human actions but are not specific to classroom contexts. These were employed to evaluate the classification performance of the Vision Mamba + APN model in complex, unconstrained settings. Second, to illustrate the potential for educational behavior modeling, a custom dataset was constructed consisting of 400 short classroom-style videos featuring 9 students performing 7 scripted actions relevant to learning scenarios (e.g., sitting, writing, raising hands, lying on the table, looking

around, playing with mobile phones). Additionally, we annotated facial expressions (e.g., anger, happiness, neutrality) using a lightweight 3D ConvNet for face detection and facial state classification. These annotations informed the development of the Teaching Evaluation Model (TEM) described in Section 5.3, where behaviors are mapped to engagement labels (positive, neutral, negative). It is important to note that the VMAPN framework itself was used for video-based behavior classification in both the standard action recognition datasets and our classroom video dataset. The 3D ConvNet was employed only as a preprocessing module for extracting facial expression labels, which serve as supplementary indicators in TEM. This separation ensures that facial state detection does not confound the evaluation of the VMAPN architecture on video action classification.

While the custom dataset offers controlled annotation and behavior consistency, it is limited in both size (400 videos) and participant diversity (9 students). This raises concerns about potential overfitting, particularly individual motion styles, clothing, or classroom layout. To mitigate this, we applied data augmentation techniques during training, including random horizontal flipping, spatial cropping, brightness jittering, and frame sequence shuffling. These augmentations improve model robustness to viewpoint and appearance variability. Despite these measures, we acknowledge that generalization remains constrained. In future work, we plan to extend evaluation to larger public datasets such as HACS, EduNet, or HMDB51, and to collect additional data across multiple institutions, age groups, and cultural settings. Such expansion will help ensure the model's scalability and reliability in real-world educational environments.

5.2 Evaluation indicators

This study follows their evaluation criteria to test the mean predictive accuracy (m AP) for different thresholds (IoU). In the THUMOS 2014 dataset, the required IoU thresholds to be tested are {0.1, 0.2, 0.3, 0.4, 0.5}; while in the ActivityNet v1.3 dataset, the required IoU thresholds to be tested are {0.5, 0.75, 0.95}. m AP averages can be used for comparing the performances between the different methods, and the resulting average prediction accuracy is used to compare the experimental results of different methods.

$$\text{mAP} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \text{AP}_t$$

$Where\ \mathcal{T}$ is the set of tested IoU thresholds, AP_t is the Average Precision computed at IoU threshold t, $|\mathcal{T}|$ is the number of thresholds in set $\mathcal{T}$. The accuracies of VMAPN and other algorithms on THUMOS 2014 dataset and ActivityNet v1.3 dataset are shown in Fig. 5, Fig. 6 respectively.

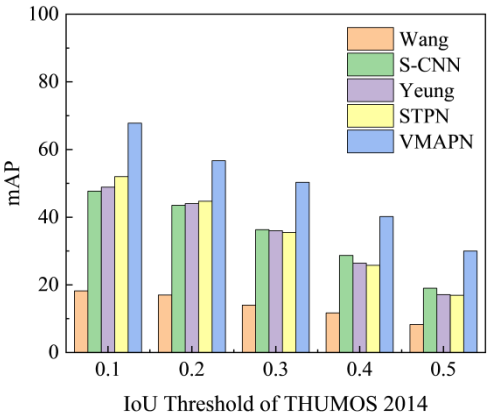

Figure 5: Accuracy of different algorithms on THUMOS 2014 dataset

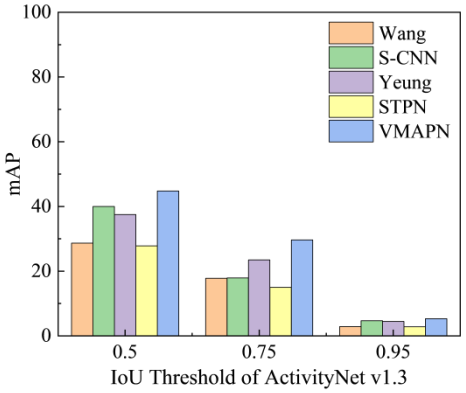

Figure 6: Accuracy of different algorithms on ActivityNet v1.3 dataset

To contextualize the effectiveness of the VMAPN framework, we compared its performance against several state-of-the-art video recognition models, including I3D, SlowFast, and TimeSformer. These models were evaluated under the same training and test conditions using the THUMOS 2014 and ActivityNet v1.3 datasets. As shown in Table 2, VMAPN achieved superior mAP scores in small-sample settings, particularly at lower IoU thresholds, demonstrating its robustness and efficiency in limited-data educational contexts.

Table 2: Performance Comparison with SOTA video recognition models

Model	Architecture Type	THUMOS 2014 (mAP@0.5)	ActivityNet v1.3 (mAP@0.5)
I3D [10]	3D CNN	58.10%	67.50%
SlowFast [12]	Dual-pathway CNN	61.90%	70.10%
TimeSformer [13]	Transformer	63.50%	70.90%
VMAPN (Ours)	ViM + APN	64.70%	72.30%

5.3 Evaluation of teaching effectiveness based on student learning behavior analysis

Traditional classroom teaching evaluations often rely on composite subjective scores from multiple sources—students, instructors, peer observers, and institutional leaders (as detailed in Table 2). While comprehensive in theory, such evaluations are inherently limited by subjectivity, inconsistency, and variability across observers. To address these limitations, this study introduces a Teaching Evaluation Model (TEM) grounded in the automated analysis of student behavioral data captured through VMAPN.

As discussed earlier, the VMAPN system detects and classifies student behaviors during classroom sessions using video-based spatio-temporal recognition. Initially, behaviors were categorized into two broad groups: positive (e.g., sitting upright, writing, raising hands, standing) and negative (e.g., lying on the table, looking away, using mobile phones). However, to improve granularity and pedagogical relevance, the TEM expands this into four behavior categories based on observational research and alignment with engagement theory: Positive behaviors (e.g., writing, raising hands, standing); Relatively positive behaviors (e.g., smiling, facing forward, communicating with the teacher); Neutral behaviors (e.g., lowering or lifting the head, sitting quietly); Negative behaviors (e.g., lying on the table, looking around, using phones). These categories form the basis for calculating an objective, behavior-informed teaching effectiveness score. To bridge the gap between VMAPN's coarse two-category output and TEM's four refined categories, we apply a mapping strategy combining VMAPN predictions with additional cues. Specifically, VMAPN identifies observable physical actions (e.g., writing, standing, lying on table) and classifies them as positive or negative learning behaviors. These predictions are then mapped into TEM's categories using rule-based logic: Actions classified as "positive" by VMAPN (e.g., writing, standing) are mapped to positive or relatively positive categories in TEM depending on facial expression cues (e.g., smiling → relatively positive). Actions such as “sitting” may be mapped to neutral if the student is inactive but not off-task. Behaviors like “looking around” or “using a phone” remain negative across both models. The scoring thresholds (e.g., 1.5–2.0 = "Good") are informed by empirical observation and cross-referenced with expert human ratings during a pilot evaluation. Table 4 summarizes the behavioral indicators and their corresponding weight within the TEM framework. This model aims to supplement traditional evaluations by offering a more objective, scalable, and consistent assessment mechanism.

Table 3: Multi-subjective evaluation indicators

First-level Indicator	Weight	Second-level Indicator	Proportion
Teaching attitude	0.15	Take the course seriously and be familiar with the content to be taught.	0.6
		Be well behaved and generous.	0.4
Teaching content	0.4	The content of the lecture strictly meets the requirements, highlighting the key points and difficulties.	0.4
		Have clear goals and there are no errors in the content.	0.3
		The content of the lecture meets the requirements and is good at combining with reality.	0.2
		Appropriately cite relevant literature.	0.1
Teaching method	0.15	Carry out education according to the textbook so that students can have a clear understanding.	0.15
		Focus on inspiration and lead students to think independently.	0.3
		The content of the class is written neatly and is good at using multimedia to improve the effect of the class.	0.3
		The lesson preparation notes are complete, the content is related to reality, and can reflect the latest scientific and technological research results.	0.25
Interactive method	0.3	Be good at mobilizing students' enthusiasm and livening up the classroom atmosphere during class.	0.6
		There are more than 5 classroom interactions in each class.	0.4
Conclusion of teaching effect evaluation >0.85: Excellent, 0.75~0.85: Good, 0.6~0.75: Qualified, <0.6:			
Unqualified mapping and quantification: 2: Excellent, 1: Good, 0: Qualified, -1: Unqualified			

Table 4: Indicators for analyzing student learning behavior

No.	Learning Behavior Classification	Corresponding Actions
1	Positive behavior	Writing, raising hands, standing up
2	Relatively positive behavior	Smiling, focusing on the front, communicating with teachers
3	Neutral behavior	Lowering head, raising head, sitting upright
4	Negative behavior	Lying on the table, looking around, playing with mobile phones
Teaching effect indicators: Good classroom effect: Good classroom effect: 1.5~2; Neutral classroom: 1~1.5; Poor classroom discipline: <1		Quantitative processing: Good classroom effect: 2; Good classroom effect: 1; Neutral classroom: 0; Poor classroom discipline: -1

$$N_p = \sum_{k=1}^{F} f_p(k),\ p = 1,2,3,4$$

The number of times of all learning behavior indicators N_p, corresponding to positive learning behaviors, more positive behaviors, neutral behaviors and negative learning behaviors, are counted according to Eq. Where $f_p(k)$ denotes the number of people in the Kth image corresponding to the indicator of learning behavior, $\sum_{k=1}^{F} f_p(k) = S$.

To reduce subjectivity in behavior labeling, we developed a standardized annotation guide used by three independent annotators with experience in educational psychology. Disagreements were resolved via majority vote.

Table 5: Per-class precision, recall, and F1-score for learning behavior recognition

Behavior	Precision	Recall	F1-Score
Sitting	0.91	0.93	0.92
Writing	0.88	0.85	0.86
Raising Hands	0.87	0.83	0.85
Standing Up	0.89	0.88	0.88
Lying on Table	0.81	0.78	0.79
Looking Around	0.76	0.72	0.74
Using Mobile Phone	0.82	0.77	0.79

Table 6: Per-class performance on facial expression recognition (LBREM)

Expression	Precision	Recall	F1-Score
Angry	0.75	0.71	0.73
Disgust	0.72	0.69	0.7
Fear	0.7	0.66	0.68
Happy	0.86	0.88	0.87
Sad	0.73	0.7	0.71
Surprise	0.8	0.81	0.8
Neutral	0.89	0.9	0.89

To compute the final \textbf{classroom effectiveness score}, we first normalize the total count of each behavior category across the observation period. Let:
N_{pos}: total count of positive behaviors
N_{rpos}: total count of relatively positive behaviors
N_{neu}: total count of neutral behaviors
N_{neg}: total count of negative behaviors
The total number of observed behaviors is given by:

$$T = N_{\text{pos}} + N_{\text{rpos}} + N_{\text{neu}} + N_{\text{neg}}$$

We then assign weights to each category:
Positive: +2
Relatively Positive: +1
Neutral: 0
Negative: -1
The Composite Behavior Score (CBS) is computed as:

$$\text{CBS} = \frac{1}{T}\left(2 \cdot N_{\text{pos}} + 1 \cdot N_{\text{rpos}} - 1 \cdot N_{\text{neg}}\right)$$

Based on the computed CBS, the classroom teaching effectiveness is categorized as:

$$\{CBS\} \geq 1.5\ \& \rightarrow \{Good\ (score = 2)\}$$
$$1.0 \leq \{CBS\} < 1.5\ \& \rightarrow \{Moderate\ (score = 1)\}$$
$$0 \leq \{CBS\} < 1.0\ \& \rightarrow \{Neutral\ (score = 0)\}$$
$$\{CBS\} < 0\ \& \rightarrow "\{Poor\ (score = -1)\}$$

This rule-based aggregation enables the transformation of frame-level behavioral observations into interpretable, high-level teaching effectiveness scores that are both quantitative and pedagogically grounded.

5.4 Ablation studies

To evaluate the contribution of each component in the VMAPN framework, we conducted a series of ablation experiments involving three modified configurations: (a) Vision Mamba (ViM) without the Augmented Prototypical Network (APN), where only the global category-labeled feature was used for classification; (b) APN applied to a Vision Transformer (ViT) backbone in place of ViM, testing the effectiveness of the prototype-based classifier independently of the ViM architecture; and (c) a baseline ViM model using only the global token for classification, without incorporating local block features or APN-based aggregation. These configurations were evaluated on the THUMOS 2014 and ActivityNet v1.3 datasets under consistent experimental conditions.

Table 7: Ablation study results on behavior classification

Model Variant	Backbone	THUMOS 2014 (mAP@0.5)	ActivityNet v1.3 (mAP@0.5)
(a) ViM only (no APN)	Vision Mamba	60.30%	68.50%
(b) APN + ViT	Vision Transformer	61.80%	69.10%
(c) Vanilla ViM (global token only)	Vision Mamba	59.40%	67.90%
(d) VMAPN (Ours)	Vision Mamba + APN	64.70%	72.30%

As shown in Table 7, the full VMAPN model outperformed all ablated variants, achieving mAP scores of 64.7% (THUMOS) and 72.3% (ActivityNet). Removing APN (variant a) led to a drop in performance to 60.3% and 68.5%, respectively, while omitting local feature integration in ViM (variant c) further reduced accuracy. Although applying APN to a ViT backbone (variant b) yielded modest improvements over vanilla ViM, it still lagged behind the VMAPN configuration. These results highlight the synergistic contribution of both Vision Mamba's spatio-temporal modeling and APN's fine-grained local feature weighting in enhancing small-sample behavior classification.

6 Conclusion

This study presents a real-time automatic scoring and feedback system for intelligent educational platforms, powered by the Vision Mamba architecture and enhanced through an Augmented Prototypical Network (APN). By leveraging spatio-temporal behavior recognition and lightweight deep learning frameworks, the proposed system effectively bridges the gap between traditional expert-based evaluation and modern AI-driven educational assessment.

Through the integration of classroom video analysis, facial expression recognition, and student behavior modeling, the system enables objective, scalable, and personalized evaluation of teaching effectiveness. Experimental validation on THUMOS 2014 and ActivityNet v1.3 demonstrates the system's high accuracy and robustness across varying IoU thresholds. In addition, the application of VMAPN in real classroom scenarios highlights its capability to capture subtle learning behaviors and translate them into meaningful feedback, both for students and instructors. Furthermore, the proposed Teaching Evaluation Model (TEM), based on student behavior classification, offers a novel perspective for educational quality assurance. It transitions from multi-subjective assessments to data-informed, real-time feedback loops, significantly enhancing the fairness and timeliness of instructional evaluation.

However, despite these promising results, the system presents several limitations that must be acknowledged. First, the reliance on facial expressions and upper-body posture may limit robustness in diverse classroom environments, particularly under poor lighting, occlusion, or cultural variation in nonverbal behavior. Second, while Vision Mamba improves efficiency, scalability remains a concern in large classrooms or multi-class deployments, where real-time processing could strain computational resources. Third, the model's performance may degrade over time due to changes in classroom dynamics or behavior patterns, highlighting the need for ongoing retraining to mitigate potential model drift.

In conclusion, this research introduces a practical and scalable AI-based system that enhances real-time educational feedback, supports small-sample learning, and aligns with national education informatization goals of smart education. While some generalizability and deployment challenges remain, this work lays the foundation for future developments in real-time educational analytics, intelligent tutoring, and adaptive teaching strategies. We also recognize that the use of classroom video data introduces important ethical considerations. To address privacy and surveillance concerns, our system follows strict data governance practices, including informed consent, secure storage, anonymization, and opt-in participation. Feedback is delivered privately to users and designed to support learning and teaching—not punitive monitoring. These safeguards aim to ensure responsible and transparent deployment in real-world educational settings. Future work will explore extending this framework to broader educational contexts, incorporating multi-modal data streams, optimizing deployment for large-scale educational platforms, and adapting the system for multi-camera classrooms and edge-device inference to enable scalable, low-latency, and infrastructure-efficient deployment in real-world settings.

Funding

1.Practical Exploration of Infusing Financial Literacy Education into Innovation and Entrepreneurship Education in Local Universities in the New Era, Higher Education Talent Development Quality and Teaching Reform Project of Sichuan Provincial Department of Education, Grant Number: JG2021-1276
2.AI-Enabled Reform and Practice of Classroom Teaching Models in Vocational Education, Chongqing Municipal Vocational Education and Teaching Reform Research Project,Grant Number: Z2252062.

References

[1] Sajja, Ramteja, et al. "Integrating AI and learning analytics for data-driven pedagogical decisions and personalized interventions in education." Technology, knowledge and learning (2025): 1-31, doi: 10.1007/s10758-025-09897-9.

[2] Sharma, Vijeta, et al. "Video processing using deep learning techniques: A systematic literature review." IEEE Access 9 (2021): 139489-139507 , doi: 10.1109/ACCESS.2021.3118541.

[3] Dai, Zhicheng, et al. "The effect of smart classrooms on project-based learning: A study based on video interaction analysis." Journal of Science Education and Technology 32.6 (2023): 858-871, doi: 10.1007/s10956-023-10056-x.

[4] Yang, Junfeng, et al. "Evaluation of smart classroom from the perspective of infusing technology into pedagogy." Smart Learning Environments 5 (2018): 1-11, doi: 10.1186/s40561-018-0070-1.

[5] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, doi: 10.1109/CVPR.2016.213.

[6] [6] Yang, Hao, et al. "Asymmetric 3d convolutional neural networks for action recognition." Pattern recognition 85 (2019): 1-12. doi:10.1016/j.patcog.2018.07.026.

[7] Bandi, Chaitanya, and Ulrike Thomas. "Skeleton-based action recognition for human-robot interaction using self-attention mechanism." 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021). IEEE, 2021. doi:10.1109/FG52635.2021.9667005.

[8] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. doi:10.1109/CVPR.2016.213.

[9] Ji, Shuiwang, et al. "3D convolutional neural networks for human action recognition." IEEE transactions on pattern analysis and machine intelligence 35.1 (2012): 221-231. doi:10.1109/TPAMI.2012.59.

[10] Wang, Yunfei, et al. "E3D: An efficient 3D CNN for the recognition of dairy cow's basic motion behavior." Computers and Electronics in Agriculture 205 (2023): 107607. doi:10.1016/j.compag.2023.107607.

[11] Kondratyuk, Dan, et al. "Movinets: Mobile video networks for efficient video recognition." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. doi:10.1109/CVPR46437.2021.01576.

[12] Wang, Limin, et al. "Tdn: Temporal difference networks for efficient action recognition." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. doi:10.1109/CVPR46437.2021.00190.

[13] Ulhaq, Anwaar, et al. "Vision transformers for action recognition: A survey." arXiv preprint arXiv:2209.05700 (2022). doi:10.48550/arXiv.2209.05700.

[14] Li, Kunchang, et al. "Uniformer: Unifying convolution and self-attention for visual recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 45.10 (2023): 12581-12600. doi:10.1109/TPAMI.2023.3274944.

[15] Kline, Michelle Ann. "TEACH: An ethogram-based method to observe and record teaching behavior." Field Methods 29.3 (2017): 205-220. doi:10.1177/1525822X16677334.

[16] Alexiadis, Dimitrios S., et al. "Evaluating a dancer's performance using kinect-based skeleton tracking." Proceedings of the 19th ACM international conference on Multimedia. 2011. doi:10.1145/2072298.2072403.

[17] Moyo, Reuben, et al. "A Video-based Detector for Suspicious Activity in Examination with OpenPose." arXiv preprint arXiv:2307.11413 (2023). doi:10.48550/arXiv.2307.11413.

[18] Zhou, Jie, et al. "Classroom learning status assessment based on deep learning." Mathematical Problems in Engineering 2022.1 (2022): 7049458. doi:10.1155/2022/7049458.

[19] Kim, Woojoo, et al. "Ergonomic postural assessment using a new open-source human pose estimation technology (OpenPose)." International Journal of Industrial Ergonomics 84 (2021): 103164. doi:10.1016/j.ergon.2021.103164.

[20] Zheng, Ce, et al. "Deep learning-based human pose estimation: A survey." ACM Computing Surveys 56.1 (2023): 1-37. doi:10.1145/3603624.

[21] Cheng, Yu, et al. "3d human pose estimation using spatio-temporal networks with explicit occlusion training." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020. doi:10.1609/aaai.v34i07.666.

[22] Xu, Rui, et al. "Visual mamba: A survey and new outlooks." arXiv preprint arXiv:2404.18861 (2024). doi:10.48550/arXiv.2404.18861.

[23] Xu, Yufei, et al. "Vitpose: Simple vision transformer baselines for human pose estimation." Advances in neural information processing systems 35 (2022): 38571-38584. doi:10.48550/arXiv.2204.12484.

[24] Idrees, Haroon, et al. "The thumos challenge on action recognition for videos “in the wild”." Computer Vision and Image Understanding 155 (2017): 1-23. doi:10.1016/j.cviu.2016.10.018.

[25] Xiong, Yuanjun, et al. "Cuhk & ethz & siat submission to activitynet challenge 2016." arXiv preprint arXiv:1608.00797 (2016). doi:10.48550/arXiv.1608.00797.

[26] Navarrete, Evelyn, et al. "A closer look into recent video-based learning research: A comprehensive review of video characteristics, tools, technologies, and learning effectiveness." International Journal of Artificial Intelligence in Education (2025): 1-64. doi:10.1007/s40593-024-00435-z.

[27] Cosentino, Giulia, et al. "Generative AI and multimodal data for educational feedback: Insights from embodied math learning." British Journal of Educational Technology (2025). doi:10.1111/bjet.13567.