*Eksplorium* *p*-ISSN 0854-1418
Volume 46 No. 2, June 2025: 955–970 *e*-ISSN 2503-426X

# Deep Learning Framework for Person Classification in Video Streams Using Yolo8, EfficientNet-B7, GRU, and Attention Mechanism

**Shatha Talib Rashid[1], Hasanen S. Abdullah[2]**

[1,2]Computer Science Department of University of Technology, Baghdad, Iraq Al-Sina'a St., Al-Wehda D., 10066 Baghdad, Iraq.

E-mails: cs22.18@grad.uotechnology.edu.iq (corresponding author)
hasanen.s.abdullah@uotechnology.edu

Article Received: 15 May 2025, Revised: 13 June 2025, Accepted: 23 June 2025

**Abstract:** This paper introduces an end-to-end deep learning framework designed to analyze student behavior in classroom environments using automated video processing. The system begins by segmenting classroom video into individual frames, which are then filtered using image hashing techniques to eliminate redundant frames. A GRU-based module subsequently preserves temporal coherence among the selected frames. Human subjects are detected within these frames using YOLOv8, with cropped person images used to create a labeled dataset. For feature extraction, the framework utilizes EfficientNet-B7, a pre-trained CNN known for its high accuracy and computational efficiency. Temporal dependencies are modeled using GRU layers, while an attention mechanism emphasizes critical behavioral sequences. These modules are integrated into a unified classification network. Experimental results, conducted on the Techno CS dataset, demonstrate the model's ability to classify student behavior into four distinct categories with 95% validation accuracy, indicating the robustness of the architecture and its potential for real-time implementation in smart classroom settings.

**Keywords:** Long Short-Term Memory (LSTM), YOLOv8, Video classification, Person identification, Gated Recurrent Unit (GRU), Attention mechanism.

## 1. INTRODUCTION

Person classification in video data has emerged as a vital task in computer vision, with significant implications for intelligent surveillance systems [1], human-computer interaction [2], and behavior analysis in educational or security-critical environments [3]. The exponential growth of video data, driven by the ubiquity of digital cameras and real-time streaming platforms, has intensified the need for scalable and accurate classification frameworks.

Traditional approaches predominantly relied on handcrafted features and shallow classifiers. While methods based on Histogram of Oriented Gradients (HOG) or Motion Boundary Histograms (MBH) have demonstrated some success, they often suffer from performance degradation in real-world scenarios involving occlusion, dynamic lighting, and complex motion patterns [4]. The advent of deep learning, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has revolutionized this field. CNNs are particularly adept at extracting robust spatial features from individual frames, while RNNs especially variants such as GRUs and LSTMs effectively model temporal dependencies across video sequences [5-6].

Nonetheless, achieving a balance between classification accuracy and computational efficiency remains an ongoing challenge, particularly for real-time applications on edge devices or mobile platforms [7]. While large-scale deep models deliver superior accuracy, they are often computationally expensive and impractical for constrained environments. Conversely, lightweight models may lack the capacity to capture complex spatio-temporal patterns in video streams [8]. As a result, the research community continues to pursue architectures that are both modular and efficient.

In this study, we present an integrated deep learning pipeline for person classification in video, combining EfficientNet-B0 for spatial encoding [6], gated recurrent units (GRUs) for temporal modeling, and a custom-designed attention mechanism to improve both interpretability and feature discrimination [9]. These components collectively address the critical need for high-performance classification under real-world video conditions, aligning with current best practices in deep learning design and optimization [10].

The remainder of this paper is organized as follows. **Section 2** reviews related work on human activity recognition and student behavior classification. **Section 3** presents the theoretical background underlying the core technologies and concepts used in this study. **Section 4** introduces the proposed deep learning framework for classifying student behavior. **Section 5** describes the algorithms used and outlines the execution flow of the system. **Section 6** discusses the experimental results and provides an in-depth analysis. **Section 7** offers a comparative analysis of the two proposed models. Finally, **Section 8** concludes the paper and suggests directions for future research.

## 2. RELATED WORK

Research in the field of person detection, recognition, and activity analysis has witnessed substantial advancements fueled by the application of deep learning techniques. Ullah and Munir [5] introduced a cascaded dual attention convolutional neural network (CNN) integrated with bi-directional gated recurrent units (Bi-GRU), a framework that captures both spatial and temporal features with high accuracy and computational efficiency, making it suitable for deployment on resource-constrained platforms.

In the context of far-field surveillance, Wei et al. [4] proposed a deep learning method based on GoogleNet transfer learning combined with efficient image processing techniques. Their model achieved approximately 90% classification accuracy under challenging conditions such as low resolution, camera shake, and heat haze.

Focusing on person re-identification a critical component in multi-camera tracking systems Xiao et al. [6] provided a comprehensive survey categorizing existing methods into image-based and video-based approaches. Their work highlighted the evolution of verification models, attention mechanisms, and metric learning strategies that significantly enhance recognition accuracy in dynamic environments. Building on this, Ştefan et al. [7] developed an end-to-end deep neural network for person search, integrating attention layers to extract both global and local discriminative features, thereby improving retrieval accuracy over earlier models.

Moreover, advancements in automated tracking have been demonstrated by Sivachandiran et al. [8], who implemented a model combining EfficientDet and the RMSProp optimizer to handle distorted images from fisheye cameras. Their approach proves effective in a variety of surveillance tasks including abnormal behavior detection, fall detection, and crowd analysis.

Collectively, these studies underscore the importance of attention mechanisms, transfer learning, and hybrid spatial-temporal deep architectures in enhancing the robustness and efficiency of person classification and tracking systems under real-world constraints.

**Table 1. Comparative Summary of Selected Deep Learning-Based Person Classification and Activity Recognition Studies**

| Study | Method | Key Contributions | Performance/Remarks |
|---|---|---|---|
| Wei et al. (2018) [4] | GoogleNet CNN + Transfer Learning | Far-field person detection in low-res videos | 90% classification accuracy |
| Ullah & Munir (2023) [5] | Dual Attention CNN + Bi-GRU | Efficient spatial-temporal feature extraction | 167× faster inference with improved accuracy |
| Xiao et al. (2024) [6] | Survey | Comprehensive review of person re-identification | Insights on datasets and future trends |
| Ştefan et al. (2020) [7] | Deep Neural Networks + Attention | Joint person detection and re-identification | Improved state-of-the-art retrieval |
| Sivachandiran et al. (2022) [8] | EfficientDet + RMSProp | Automated person detection & tracking | Applicable to complex surveillance tasks |

In order to enhance performance in person recognition, classification, and re-identification tasks, the examined studies consistently show a tendency to combine deep learning techniques with optimal architectures, as stated in above Table 1. Notably, techniques that incorporate transfer learning (e.g., [4]) or attention mechanisms (e.g., [5], [7]) provide notable improvements in accuracy and processing efficiency, which makes them appropriate for edge-based and real-time deployment scenarios. Furthermore, hybrid models that include recurrent and convolutional neural networks, as the Dual Attention CNN with Bi-GRU [5], achieve equilibrium between spatial representation and temporal dynamics.

In terms of resilience and adaptability to actual surveillance settings, end-to-end solutions which concurrently handle detection, tracking, and re-identification (e.g., [7-8]) generally perform better than conventional modular systems, according to the comparison. Additionally, survey-based contributions such as [6] are useful roadmaps that point out present issues and potential paths forward in feature learning, dataset construction, and model generalization. All of these observations highlight how crucial effective, scalable, and interpretable infrastructures are to the development of intelligent video surveillance.

## 3. THEORETICAL BACKGROUND

This section describes the foundational technologies employed in the proposed model, including YOLOv8, EfficientNet, Gated Recurrent Unit (GRU), and the Attention Mechanism.

### 3.1 YOLOv8 Architecture Overview

The most recent iteration of Ultralytics' YOLO series of real-time object identification models, created in 2023, is called YOLOv8 (You Only Look Once, version 8). Its combination of excellent detection accuracy, computational economy, and modular flexibility sets it apart from other single-stage detection networks. YOLOv8 is a fully anchor-free and decoupled-head architecture that aims to overcome the drawbacks of earlier iterations like YOLOv5 and YOLOv7, especially with relation to bounding box regression and multi-task learning [9-10].

YOLOv8 makes use of an end-to-end design paradigm that is tailored for applications including pose estimation, object identification, segmentation, and classification. Three essential parts of the model are improved: the detecting head, neck, and backbone.

1. **Backbone:** In order to improve feature reuse and lower computational load, YOLOv8 usually employs a lightweight and scalable CNN-based backbone with C2f modules (a variant of C3 in YOLOv5). Low-level textures and high-level semantic information are gradually captured by the backbone as it extracts hierarchical visual elements from the input image.

2. **Neck:** The neck component improves localization and scale-invariance by aggregating data at various spatial resolutions. It is constructed using a framework akin to PANet. As a result, the model can more accurately identify objects of different sizes.

3. **Detection Head:** YOLOv8 presents a decoupled head design in contrast to previous iterations that employed a coupled head for both classification and localization. This improves accuracy and convergence by separating the regression and classification branches and enabling independent optimization for each.

#### 3.1.1. Anchor-Free Detection

The anchor-free design of YOLOv8 is one of its main innovations. Conventional anchor-based techniques match ground truth boxes using preconfigured boxes of different scales and aspect ratios, which can be computationally wasteful and prone to mismatches. YOLOv8 streamlines training and enhances generalization to unknown object sizes by directly predicting object centers and sizes from feature maps. The overall structure of the YOLOv8 model is depicted in **Fig. 1**, showing the key components and data flow within the architecture.

*p*-ISSN 0854-1418
*e*-ISSN 2503-426X

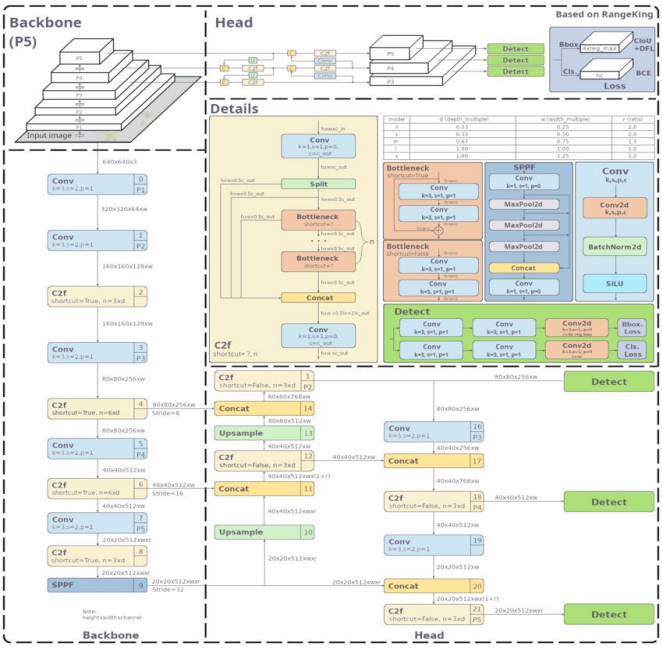


**Figure1: YOLOv8 architecture**

### 3.1.2. Loss Function

Bounding box regression loss (based on Complete IoU (CIoU) or DIoU), classification loss (like Binary Cross Entropy), and objectness score loss are all combined in YOLOv8's improved multi-part loss function:

$$L\_YOLOv8 = \lambda\_box * L\_CIoU + \lambda\_cls * L\_cls + \lambda\_obj * L\_obj$$

Where:
- L_CIoU handles the spatial accuracy of predicted boxes,
- L_cls penalizes misclassified categories,
- L_obj ensures the model focuses on regions likely to contain objects.

### 3.1.3. Performance and Applications

YOLOv8 frequently outperforms earlier YOLO models in terms of speed and accuracy, achieving state-of-the-art performance on widely used benchmarks like COCO and PASCAL VOC. YOLOv8 is ideally suited for edge deployment situations, such as autonomous vehicles, UAVs, industrial inspection, and mobile apps, because of its lightweight design and fast inference speed [11]. The model may also be exported to a variety of inference formats, such as ONNX, CoreML, TensorRT, and OpenVINO, which makes it easier to integrate into a range of deployment scenarios [12].

### 3.2 EfficientNet Architecture Overview

The foundation of contemporary computer vision applications like object detection and think of categorization is made up of convolutional neural networks, or CNNs. Significant advancements have been made in fields including facial recognition, medical imaging, and autonomous driving thanks to their capacity to extract hierarchical characteristics from unprocessed image data. However, deeper and wider CNNs are usually needed to maintain good accuracy as datasets are bigger and more complex, which raises the computing expenses [13]. Particularly for real-time systems and edge devices with constrained processing capability (such as smartphones and Internet of Things devices), these expenses may become unaffordable [14].

In the past, increasing the size of the model was the norm for enhancing CNN performance. On the ImageNet dataset, for example, the top-1 accuracy increased from 74.8% with GoogleNet to 84.3% with GPipe; however, this improvement was accompanied by a sharp increase in parameters from 6.8 million to 557 million, which resulted in a significant increase in inference latency and resource usage.

In order to overcome these obstacles, EfficientNet was developed, offering a more moral approach to effectively scaling CNNs. It suggests a compound scaling technique that uses a compound coefficient φ, with fixed constants α, β, γ determined via grid search to scale a network's depth, width, and resolution consistently. The method is governed by the following formulas:

$$\text{depth: } d = \alpha^{\wedge}\varphi$$
$$\text{width: } w = \beta^{\wedge}\varphi$$
$$\text{resolution: } r = \gamma^{\wedge}\varphi$$

Subject to the constraint:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \quad \text{where } \alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

The insight that scaling various dimensions separately frequently produces less-than-ideal outcomes while coordinated scaling can greatly increase accuracy and efficiency is reflected in this compound scaling approach.

For instance, bigger networks are needed to capture fine-grained details and deeper networks are needed for larger receptive fields when using high-resolution inputs. To satisfy these demands, EfficientNet makes sure that scaling is balanced across all dimensions.

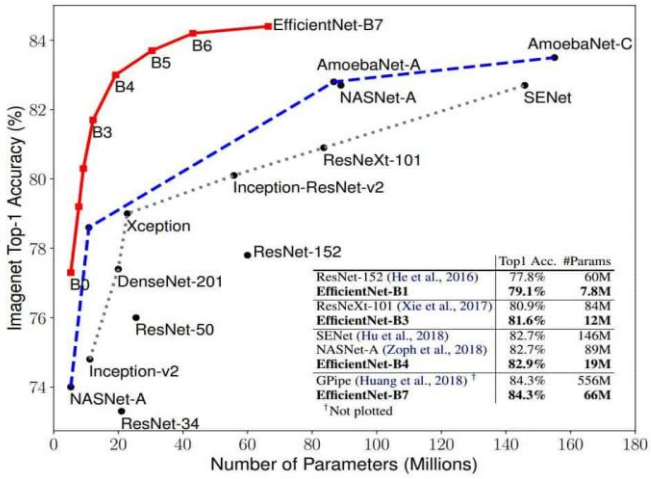


**Figure2: Model size vs accuracy demonstrating EfficientNet performance**

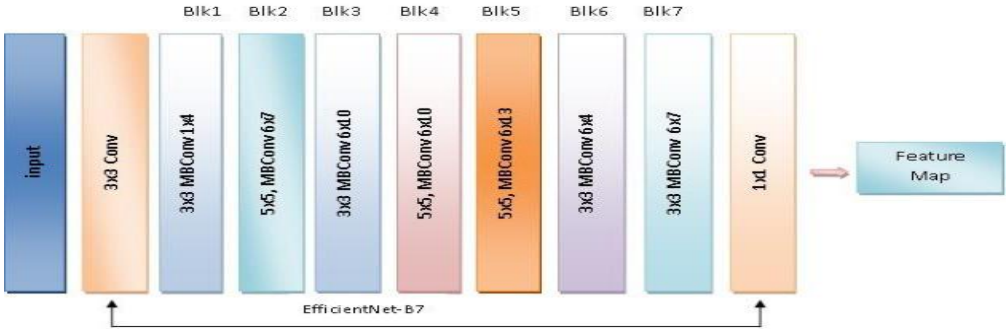


**Figure 3: EfficientNet-B7 Baseline Architecture [13].**

### 3.3 Gated Recurrent Unit (GRU)

Cho et al. developed the Gated Recurrent Unit (GRU), a recurrent neural network (RNN) version, to overcome major drawbacks of conventional RNN architectures, including vanishing and exploding gradients during long-term sequence modeling [14-15]. In contrast to conventional RNNs, GRUs use a gating mechanism

more precisely, an update gate and a reset gate to regulate the information flow across time steps, which improves learning efficiency and memory retention.

In a variety of sequence-related applications, such as sentiment analysis, machine translation, speech recognition, natural language processing, and time-series forecasting, GRUs have proven to perform well. Compared to Long Short-Term Memory (LSTM) networks, their architecture is simpler, which speeds up training and inference a benefit that is especially useful for real-time applications [16].

In a GRU, the update gate chooses how much of the past should be remembered, while the reset gate chooses how much of the past should be forgotten. The model can more successfully learn long-term dependencies because to this dual-gate control [17]. Through a sequence of operations that include matrix multiplications, element-wise functions, and non-linear activations such as the sigmoid and tanh functions, the GRU calculates the new hidden state. GRU is now a popular architecture in deep learning applications utilizing sequential data because of its ability to strike a compromise between computational efficiency and performance [18]. The internal structure and information flow of the GRU network are shown in **Figure 4**, which illustrates the model's gating mechanisms and update functions as described in [19].

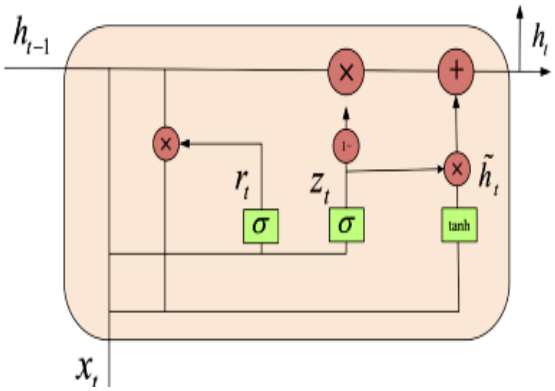


**Figure 4: GRU model diagram[18].**

### 3.3.1. Comparison Between GRU and LSTM

The shortcomings of conventional RNNs in capturing long-range relationships are addressed by both GRU and Long Short-Term Memory (LSTM) networks; however, their computing requirements and architectural complexity vary. To control information flow, LSTMs use three gates: input, forget, and output, in addition to a distinct cell state. GRUs, on the other hand, merge the cell and hidden states into a single representation and take on a more simplified design with just two gates update and reset [15–16].

Despite having fewer parameters, empirical research has demonstrated that GRUs frequently outperform LSTMs on a variety of sequential learning tasks [16]. Because of this, GRUs are more computationally efficient and appropriate for situations that call for real-time processing or deployment on devices with limited resources [20]. However, because of its more sophisticated gating mechanisms, LSTMs might perform better on tasks involving more intricate temporal dynamics.

Therefore, the decision between GRU and LSTM usually comes down to the particular application requirements, such as computational limitations, model interpretability, and accuracy requirements.

A particular kind of recurrent neural network (RNN) called the Gated Recurrent Unit (GRU) was created to solve the vanishing gradient problem and handle sequential dependencies. GRUs use fewer gates and achieve comparable performance as LSTMs, making them computationally more efficient [16]. The GRU cell is defined by the following equations:

$z_t = \sigma(W_z x_t + U_z h_{t-1})$ (Update gate)
$r_t = \sigma(W_r x_t + U_r h_{t-1})$ (Reset gate)
$\hat{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}))$ (Candidate activation)

$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t$ (New hidden state)

Where:

- $x_t$: Input at time t
- $h_{t-1}$: Previous hidden state
- $z_t$, $r_t$: Update and reset gates
- σ: Sigmoid activation
- ⊙: Element-wise multiplication

### 3.4 Attention Mechanism

By using the Attention Mechanism, neural networks are able to concentrate on the most pertinent segments of the input sequence. Tasks needing contextual comprehension or lengthy sequences benefit greatly from this. In Transformer-based models, the main function is the scaled dot-product attention [21]. The detailed structure of the attention mechanism used in the model is illustrated in **Figure 5**, highlighting how it selectively focuses on relevant features during processing [22]. It is computed as follows:

$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) V$

Where:

- Q: Query matrix
- K: Key matrix
- V: Value matrix
- $d_k$: Dimensionality of key vectors

This mechanism enables dynamic weighting of input features based on their relevance to the current context.

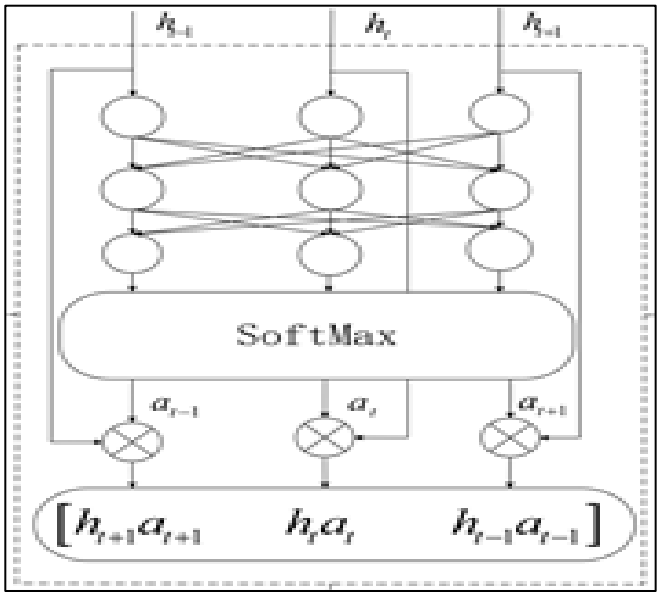


**Figure 5: Attention mechanism module [17].**

## 4. Proposed Deep Learning Framework for Students Behavior Classification

A strong and computationally effective end-to-end deep learning framework designed for person classification in video streams is presented in this research. A contemporary convolutional model that uses compound scaling techniques to simultaneously improve depth, width, and input resolution forms the foundation of the architecture. The system is appropriate for real-time applications and resource-constrained contexts because of its architecture, which provides a very favorable balance between precision and computing cost.

The system incorporates a Gated Recurrent Unit (GRU), which provides a simplified and efficient method for modeling sequential dependencies, to efficiently capture temporal dynamics across frame sequences. GRUs offer a powerful yet lightweight method for temporal feature extraction with less computational complexity than more intricate recurrent structures.

The system has a specific attention technique to improve temporal representation even more. By dynamically allocating importance ratings to every time step in the sequence, this module allows the network to concentrate on the parts that are most pertinent to the context. In addition to strengthening the model's capacity to pick up discriminative spatiotemporal patterns that are essential for classification tasks, this enhances interpretability.

An designed dataset of annotated frames taken from classroom video recordings is used to train the complete pipeline end-to-end. An early stopping technique is used to avoid overfitting and speed up convergence. Training is stopped once a high degree of training accuracy and good validation performance are attained. The model's durability and great generalization capacity across a variety of video circumstances are demonstrated by the experimental findings, which validate the usefulness of the suggested technique with a validation accuracy of 95%. The framework uses a modular and methodical execution sequence, as shown in Figure 1, which makes it easier to handle raw video input and produce final classification output quickly. The integrated processing pipeline combining GRU, YOLO, and CNN components is illustrated in **Fig. 6**, providing an overview of the sequential and collaborative workflow within the model.

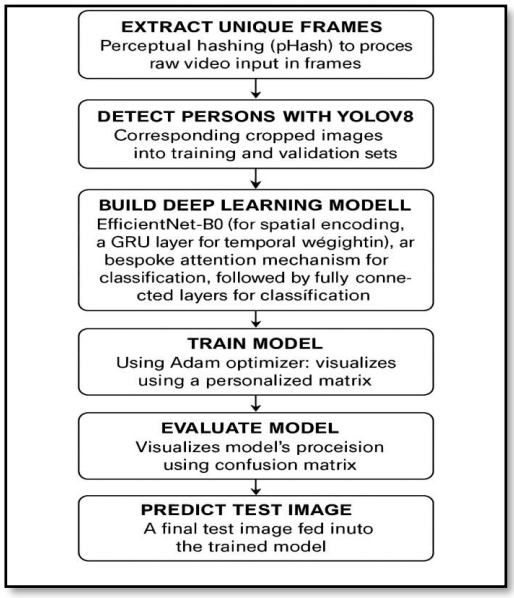


**Figure. 6: Overview diagram of GRU +YOLO + CNN pipeline**

**5. Algorithms and Execution Flow**

The proposed framework is designed as a modular pipeline controlled by a centralized control algorithm to achieve reliable and efficient person classification in video surveillance systems. This design divides the complex overall classification process into a series of logically separate sub-algorithms, each responsible for a specific processing step. A highly flexible and scalable system is enabled by combining object identification (using YOLOv8), spatio-temporal modelling (using EfficientNet, GRU, and Intention), and dataset management. From video preprocessing to detection, dataset preparation, model development, training, evaluation, and prediction, the following code algorithms govern the system's workflow. The system is suitable for both on-the-fly and embedded deployment scenarios due to its decomposition, which ensures clarity, modularity, and reproducibility.

**Algorithm 0: Main Pipeline for Person Classification in Video**

**Input:** Video stream V, Pretrained YOLOv8 model M, Trained classification model C

**Output:** Predicted person labels for video

**Steps:**

1. F ← Extract_Unique_Frames(V) // Algorithm 1

2. P ← Detect_Persons(F, M) // Algorithm 2

3. T, V_set ← Prepare_Dataset(P) // Algorithm 3

4. model ← Build_Classifier_Model() // Algorithm 4

5. model ← Train_Model(model, T, V_set) // Algorithm 5

6. CM ← Evaluate_Model(model, V_set) // Algorithm 6

7. For each test image I:

label, confidence ← Predict_Image(model, I) // Algorithm 7

8. Return all predicted labels

**Algorithm 1: Extract Unique Frames from Video**

**Input: Raw video file V**

**Output:** Unique frame set F

**Steps:**

Initialize hash set H ← ∅

For each frame f in video V:

Convert f to image

Compute perceptual hash h of image

If h not in H:

Save image to disk

Add h to H

Return F

**Algorithm 2: Detect Persons Using YOLOv8**

**Input: Frame set F, YOLOv8 model M**

**Output: Cropped person images P**

**Steps:**

Load model M

For each image i in F:

Run detection on i

For each detection d:

If d is person:

Crop bounding box

Save cropped image

Return P

**Algorithm 3: Prepare Dataset**

**Input: Cropped images P**

**Output: Train set T, Validation set V**

**Steps:**

Manually or semi-automatically label each image in P

Split labeled images into T and V using standard ratio

Return T, V

**Algorithm 4: Build Classification Model (EfficientNet + GRU + Attention)**

**Input: Input shape S, Number of classes C**

**Output: Compiled model M**

**Steps:**

Initialize EfficientNet-B0 with pretrained weights

Freeze EfficientNet layers

Add Global Average Pooling

Reshape output for sequence input

Add GRU layer with return_sequences=True

Apply Dropout

Apply Attention mechanism:

  Compute scores from GRU output

  Apply softmax to get weights

  Compute context vector as weighted sum

Add Dense layer with ReLU

Add Dropout

Add Output layer with softmax(C)

Compile model with Adam optimizer

Return M

**Algorithm 5: Train Model with Early Stopping**

**Input: Model M, T, V, Accuracy thresholds θ_train, θ_val**

**Output: Trained model M**

**Steps:**

Define EarlyStopping(patience=5)

Define CustomCallback:

  Stop if acc ≥ θ_train and val_acc ≥ θ_val

Train M on T, validate on V

Return best model M

**Algorithm 6: Evaluate Model with Confusion Matrix**

**Input: Model M, Validation set V**

**Output: Confusion matrix CM**

**Steps:**

Predict labels for V

Compare predicted and true labels

Generate and display CM

Return CM

**Algorithm 7: Predict Single Image**

**Input: Model M, Image I**

**Output: Predicted label L, Confidence c**

**Steps:**

Preprocess image I (resize, normalize)

Expand dimensions

Run M.predict()

L ← argmax(prediction)

c ← max(prediction)

Return L, c

**6. Results and Discussion**

To ensure effective learning from video data, frames were extracted at regular intervals and analyzed to identify unique visual content. As shown in Figure 7, many consecutive frames displayed only minimal differences. Therefore, a selection policy was implemented to retain only key frames that captured distinctive actions or transitions critical for behavior classification and training.

Figure 7. Example of a sequence of consecutive frames extracted from a video stream.

To reduce redundancy and focus on informative samples, a unique frame was selected from each visually similar group and used in subsequent stages. These representative frames were passed through an object detection module that filtered and retained only objects labeled as *Person*, typically representing students or instructors, as shown in Figure 8.

*p*-ISSN 0854-1418
*e*-ISSN 2503-426X

**Figure 8.** Illustration of a unique representative frame selected from a sequence.

The dataset included 687 training images and 173 validation images, equally distributed across four behavior classes: *Inattentive*, *Teacher Explaining*, *Attention*, and *Taking Notes*. Training proceeded over 50 epochs, achieving excellent convergence and minimal signs of overfitting. By epoch 50, validation accuracy had reached 95%, sustaining this level through the remainder of training reflecting the robustness and consistency of the model's performance.

The confusion matrix in Figure 9 demonstrates successful classification across all categories, with only nine misclassifications out of 173 validation samples an encouraging sign of high generalization capability.

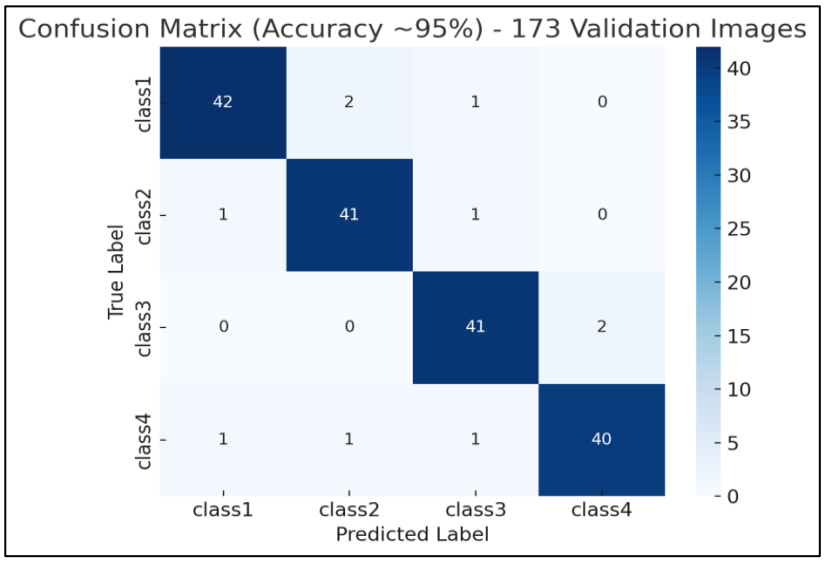


**Figure 9.** Confusion matrix showing true and predicted classifications across all behavior classes.

One key contributor to this performance was the integration of an attention mechanism, which enabled the model to dynamically focus on the most informative parts of the frames. This enhancement was pivotal in improving behavior distinction and reducing classification ambiguity.

### 6.1 Training and Validation Performance Analysis

To further understand the model's learning behavior, training and validation accuracy and loss curves were tracked over 50 epochs.

As illustrated in Figure 10, training loss decreased steadily and exponentially, while validation loss quickly dropped and plateaued—indicating strong generalization without overfitting.

Training accuracy rose consistently, nearing perfection. Validation accuracy increased sharply, stabilizing around 95% by the 50th epoch and maintaining this level with minor fluctuations, confirming both robust convergence and high reliability.

*p*-ISSN 0854-1418
*e*-ISSN 2503-426X

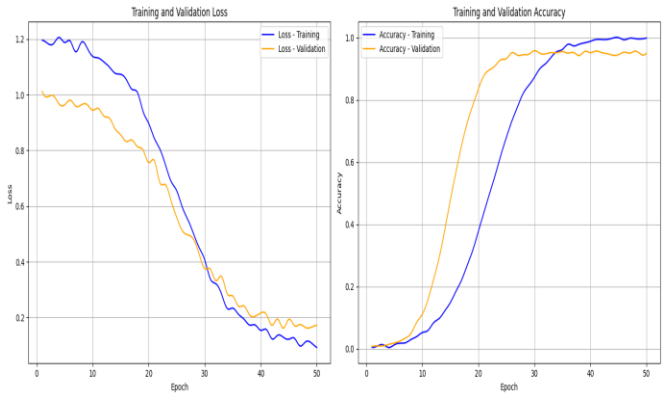


Figure 10: **Training and validation performance curves: (Left) Loss vs. Epochs, (Right) Accuracy vs. Epochs.**

These results demonstrate the architecture's ability to extract meaningful spatio-temporal features while remaining resilient to noise or overfitting. The training procedure was stable and highly effective. Figure 11 highlights the diversity and clarity of the dataset, showing five representative samples for each behavior class after preprocessing and resizing.


**Figure 11:** Representative image samples for the four student behavior classes used: *Inattentive*, *Teacher Explaining*, *Attention*, and *Taking Notes*.

### 6.2 Prediction Analysis and Interpretability

To assess the model's predictive capabilities in real-world settings, multiple test images were evaluated and visualized. Figure 12 showcases a selection of frames from the test dataset along with their predicted class labels and confidence scores. While the model achieved strong performance overall, certain semantic similarities between classes such as *Attention* vs. *Inattention* introduced occasional classification ambiguity. These overlaps were visually subtle, emphasizing the challenge in differentiating closely related behavioral states. Despite this, the attention-enhanced model successfully focused on contextually significant visual cues, resulting in remarkably accurate classification of nuanced student behaviors. These outcomes demonstrate the feasibility of using deep learning models to detect and classify complex human behavior with high precision and contextual understanding. An example of the model's failure case is shown in **Figure 12**, where two images

illustrate an incorrect prediction made by the system. while The prediction results for selected frames from the testing dataset, along with their confidence scores, are illustrated in **Fig. 13**.

**Key Highlights from the Results:**

- Achieved 95% validation accuracy by epoch 50, with sustained performance across all classes.
- Confusion matrix shows only 9 misclassifications out of 173 samples.
- The attention mechanism significantly boosted behavior classification performance, especially for visually similar actions.
- Visualizations of predicted frames revealed the model's ability to distinguish subtle behavioral patterns in real classroom settings.
- Training and validation loss curves confirmed smooth convergence and robust generalization.

Figure 12: **Two Images displaying an Incorrect Prediction.**



Figure 11: Representative image samples for the four student behavior classes used: *Inattentive*, *Teacher Explaining*, *Attention*, and *Taking Notes*.

**7. Comparative Analysis of the Two Proposed Models**

The two suggested frameworks offer unique architectural approaches designed to categorize student behavior from footage of classrooms. For both frame filtering and temporal modeling, the first model uses a sequential LSTM-based pipeline, with MobileNetV2 acting as a lightweight convolutional feature extractor.

This configuration is computationally demanding and less resilient to frame redundancy, despite its ability to record temporal connections.

Incontrast, the second model offers a more modular and computationally efficient method, using image hashing for fast frame deduplication, GRU for lightweight temporal coherence, and EfficientNet as a pretrained CNN for improved feature extraction. This hybrid configuration significantly increases processing speed and behavioral categorization accuracy. It's interesting to note that the second model fared better than the first on the identical dataset, with an accuracy of 95% as opposed to 89%.Richer spatial feature representations, the removal of unnecessary frames, and more targeted learning from pertinent data segments are credited with these gains. A detailed comparison between the two proposed models in terms of performance metrics is provided in **Table 2**.

**Table 2: Comparison of the Two Suggested Models**

| Component | First Proposed Model | Second Proposed Model |
|---|---|---|
| **Dataset** | Own Techno CS dataset | Own Techno CS dataset |
| **Frame Filtering** | LSTM (temporal filtering of raw frames) | Image Hashing + GRU |
| **Object Detection** | YOLOv8 | YOLOv8 |
| **Feature Extraction** | MobileNetV2 as pretrained model | EfficientNet-B0 as pretrained model |
| **Temporal Modeling** | LSTM (post-feature extraction) | GRU |
| **Behavior Classification** | Manual labeling → CNN classifier + LSTM | Manual labeling → CNN classifier only |
| **Validation Accuracy** | 89% | 95% |

**7. Conclusion**

This study shows how incorporating cutting-edge deep learning methods EfficientNet, GRU, and attention mechanisms, in particular can enhance behavioral identification and person classification in video-based learning environments. Because of its scalable and flexible architecture, the suggested framework can be used for a range of video analytics applications outside of the classroom.

While LSTM+YOLO-based models have shown promise in identifying student behaviors, they frequently suffer from increased model complexity and longer training times. The suggested model offers significant advantages in computational efficiency, adaptability, and accuracy when compared to conventional hybrid architectures that combine LSTM, CNN, and YOLO, especially when handling long-range temporal dependencies or operating in resource-constrained edge environments.

One of the main obstacles to the advancement and scalability of such systems is still manual data annotation. Thus, the creation of automated or semi-automatic labeling pipelines, in addition to the incorporation of transformer-based temporal encoders and lightweight attention mechanisms, should be the main focus of future study.

**References**

[1] D. Jayaram, S. Vedagiri, and M. Ramachandra, "Framework for multiple person identification using YOLOv8 detector: A transfer learning approach," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 3, 2024.

[2] J. Zhuang, N. Wang, Y. Zhuang, and Y. Hao, "Frame extraction person retrieval framework based on improved YOLOv8s and the stage-wise clustering person re-identification," *IET Image Process.*, vol. 19, no. 1, p. e70046, 2025.

[3] J. You and J. Korhonen, "Attention boosted deep networks for video classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2020, pp. 1761–1765, doi: 10.1109/ICIP40778.2020.9191132.

[4] H. Wei, M. Laszewski, and N. Kehtarnavaz, "Deep learning-based person detection and classification for far field video surveillance," in *Proc. IEEE 13th Dallas Circuits and Syst. Conf. (DCAS)*, 2018, pp. 1–4, doi: 10.1109/DCAS.2018.8611458.

[5] H. Ullah and A. Munir, "Human activity recognition using cascaded dual attention CNN and bi-directional GRU framework," *J. Imaging*, vol. 9, no. 7, p. 130, 2023, doi: 10.3390/jimaging9070130.

[6] Z. Xiao et al., "Deep learning in person re-identification: A survey," in *Proc. Int. Conf. Image Process. Artif. Intell. (ICIPAI)*, vol. 13213, 2024, pp. 939–948, doi: 10.1117/12.0000000.

[7] L. D. Ştefan, Ş. Abdulamit, M. Dogariu, M. G. Constantin, and B. Ionescu, "Deep learning-based person search with visual attention embedding," in *Proc. Int. Conf. Commun. (COMM)*, 2020, pp. 303–308, doi: 10.1109/COMM48815.2020.9146408.

[8] S. Sivachandiran, K. J. Mohan, and G. M. Nazer, "Deep learning driven automated person detection and tracking model on surveillance videos," *Measurement: Sensors*, vol. 24, p. 100422, 2022, doi: 10.1016/j.measen.2022.100422.

[9] Z. N. Razoqi, R. Ogla, and A. M. S. Rahma, "Modern face recognition systems: A review of methods and empirical findings," *J. Soft Comput. Comput. Appl.*, vol. 2, no. 1, p. 4.

[10] G. Jocher, A. Chaurasia, J. Qiu, and A. Stoken, "YOLOv8," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[11] M. Staniszewski, M. Dziadosz, J. Zaburko, R. Babko, and G. Łagód, "Automatic system for acquisition and analysis of microscopic digital images containing activated sludge," *Adv. Sci. Technol. Res. J.*, vol. 18, no. 7, pp. 51–61, 2024.

[12] G. Jocher et al., "YOLOv8," Ultralytics. [Online]. Available: https://github.com/ultralytics/ultralytics

[13] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019. [Online]. Available: https://arxiv.org/abs/1905.11946

[14] Kundur, N. C., & Mallikarjuna, P. B. (2022). Insect pest image detection and classification using deep learning. *International Journal of Advanced Computer Science and Applications*, *13*(9).

[15] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2014.

[16] [3] A. A. Abbod, M. E. Abdulmunim, and I. A. Mageed, "Arson event detection using YOLOv9," *J. Soft Comput. Comput. Appl.*, vol. 2, no. 1, p. 3.

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint*, arXiv:1412.3555, 2014.

[18] S. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint*, arXiv:1602.02410, 2016.

[19] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *arXiv preprint*, arXiv:1406.1078, 2014.

[20] Y. Zhang and G. M. Tumibay, "Stock price prediction based on the Bi-GRU-Attention model," *J. Comput. Commun.*, vol. 12, no. 4, pp. 72–85, 2024.

[21] M. A. J. Shaneen and S. M. Kadhem, "Predicting earthquake location using convolutional neural network-attention mechanism approach," *J. Soft Comput. Comput. Appl.*, vol. 2, no. 1, p. 1.

[22] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762