International Journal of Applied Mathematics and Computing
E-ISSN: 3047-146X
P-ISSN: 3048-1988
IJAMC

Research Article

Detecting Levels of Learning Concentration Through Student Behavior in the Classroom Using Convolutional Neural Networks (CNN)

Yuma Akbar[1]*, Sopan Adrianto[2], Rasiban Rasiban[3], Nadya Khairunnisa[4]

[1]Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika Jakarta, Indonesia;
email: yumekhan@stikomcki.ac.id
[2]Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika Jakarta, Indonesia;
email: sopan@stikomcki.ac.id
[3]Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika Jakarta, Indonesia;
email: rasiban@stikomcki.ac.id
[4]Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika Jakarta, Indonesia;
email: ndyaakn@gmail.com
*Corresponding Author: yumekhan@stikomcki.ac.id

Abstract: This study discusses a student concentration detection system using Convolutional Neural Network (CNN) with the MobileNetV2 architecture. The dataset was adapted from Classroom Student Behaviors and mapped into four concentration categories: highly focused, focused, less focused, and unfocused. The system was tested with a 720p webcam and produced real-time detection data. The evaluation results show an overall accuracy of 75.85%, with the highest precision achieved in the focused class (0.9859) and the highest recall in the highly focused (0.9739) and unfocused (0.9811) classes. The confusion matrix indicates that the focused class was detected most consistently, while highly focused and unfocused classes were often misclassified as focused, resulting in lower precision. In real-time testing, the system operated at an average of 7 FPS and worked optimally when students faced the camera directly with sufficient lighting, but its performance decreased significantly at face angles greater than 45°. User evaluation shows that 75% of students rated the detection results as accurate/very accurate with an average satisfaction score of 3.6 out of 5, and 75% felt assisted in recognizing their concentration level. From the teachers' perspective, most stated that the results were consistent with classroom observations, and all expressed willingness to reuse the system.

Keywords: CNN; Computer Vision; Concentration; Learning; MobileNetV2.

Received: 20 November 2024
Revised: 16 December 2024
Accepted: 11 January 2025
Published: 30 January 2025
Curr. Ver.: 30 January 2025

1. Introduction

Concentration is the ability to focus attention on an object or activity for a certain period of time without being distracted by other factors. Learning concentration is one of the cognitive aspects that determines an individual's success in understanding the information they receive (Mudjiono, 2009). In the educational context, learning concentration is a key factor influencing how effectively students can absorb, comprehend, and process information presented during the learning process. Students with a high level of concentration tend to understand lessons more easily, actively participate in classroom activities, and complete assignments effectively. Conversely, students with low levels of concentration are more

susceptible to distractions, have difficulty following the learning process, and are at greater risk of experiencing a decline in academic performance.

However, in practice, monitoring students' learning concentration in the classroom presents several challenges. Teachers are generally able to observe only a limited number of students simultaneously, resulting in an incomplete understanding of the overall classroom condition. This challenge becomes even more significant in large classes, where teachers must divide their attention between managing classroom dynamics and monitoring individual students' concentration levels. Additional factors influencing student concentration include differences in learning styles, students' physical and psychological conditions, environmental distractions, and limited instructional time. These conditions make manual observation a less effective method for assessing learning concentration.

Furthermore, student concentration is inherently dynamic. Concentration levels can change within minutes depending on students' interest in the learning material, emotional state, and classroom environment. For example, a student may appear highly focused at the beginning of a lesson while listening to the teacher's explanation, but their concentration may decline shortly afterward due to boredom or external distractions. Such fluctuations are difficult for teachers to monitor through manual observation, often resulting in inaccurate or biased assessments.

Artificial Intelligence (AI) technology, particularly Computer Vision, offers a new approach to monitoring students' learning behaviors and facial expressions. By utilizing cameras and image classification algorithms, a system can identify specific behavioral patterns such as looking away, feeling drowsy, reading, writing, or raising a hand. This technology not only improves the efficiency of classroom observation but also creates opportunities for more personalized learning experiences.

Therefore, research is needed to develop a student learning concentration detection system based on classroom learning behaviors using a Convolutional Neural Network (CNN) approach. CNN is widely recognized for its high accuracy in image classification tasks and is well suited for application in classroom learning environments.

This study is expected to provide an innovative solution for teachers to monitor students' concentration levels in the classroom while contributing to the development of educational technology ecosystems that support more effective teaching practices in the future. Through a real-time detection system, teachers can obtain accurate data that enable timely and appropriate instructional interventions.

2. Literature Review

Learning Concentration

Learning concentration in this study is understood as students' ability to focus their attention, thoughts, and behavior on classroom learning activities. Learning concentration refers to the ability to focus attention on learning materials without being distracted by external factors (Slameto, 2010). This is consistent with the view that concentration is a crucial aspect of attention that strongly influences learning success (Winkel, 2009).

Previous studies have categorized student engagement into multiple levels. Using the DAiSEE dataset, Gupta et al. (2017) classified engagement into very low, low, high, and very high categories. Similarly, EEG-based studies by Zheng and Lu (2015) and Li et al. (2016) categorized attention into four levels: high, medium, low, and non-focused.

Based on these foundations, this study adopts four concentration categories: highly focused, focused, less focused, and unfocused. These categories are operationalized through observable classroom behaviors using the Behavior Observation of Students in Schools (BOSS) framework, which classifies student behaviors into Active Engagement, Passive Engagement, and Off-task behaviors (Hintze et al., 2002; Alperin et al., 2023).

Highly Focused

This category describes students who demonstrate full attention and active involvement in learning activities. The behavioral indicator associated with this category is raising hand. Within the BOSS framework, this behavior is classified as Active Engagement because it reflects active participation, such as asking or answering questions (Hintze et al., 2002).

Focused

This category indicates that students remain engaged with learning materials through core learning activities, although they may not actively participate in discussions. Behaviors included in this category are looking forward, writing, and reading. According to the BOSS framework, these behaviors are categorized as Passive Engagement or low-level Active

Engagement because students remain on-task while paying attention to the teacher, reading instructional materials, or taking notes (Hintze et al., 2002).

Less Focused

This category indicates that students are physically present in the classroom but their attention has begun to drift. The behaviors associated with this category are turning around and standing without instructional purposes. In the BOSS framework, these behaviors are classified as Off-task Motor behaviors because they involve physical activities unrelated to classroom tasks (Hintze et al., 2002).

Unfocused

This category indicates that students are not engaged in the learning process. The behavioral indicator associated with this category is sleeping in class. BOSS and other observation instruments categorize sleeping or resting one's head on the desk as Off-task Passive behavior (Hintze et al., 2002; Avon-Washington County Schools, 2015).

Computer Vision

Computer Vision is a branch of computer science that focuses on enabling computers to acquire, process, and interpret information from images or videos in a manner similar to human vision. The primary goal of computer vision is to allow machines to recognize and understand objects and their surrounding environment through visual data.

According to Szeliski (2022), computer vision encompasses a variety of techniques for analyzing and interpreting images and has applications in numerous real-world domains, including facial recognition, autonomous vehicle navigation, and surveillance systems. Gonzalez and Woods (2008) further explain that computer vision operates at a higher level than image processing because it not only processes visual information but also seeks to understand its meaning.

Python Programming Language

Figure 1. Python Logo.

Python is a high-level programming language designed with an emphasis on code readability and simplicity. Developed by Guido van Rossum and officially released in 1991, Python supports multiple programming paradigms, including object-oriented, imperative, and functional programming.

Python is widely recognized for its concise and expressive syntax, making it popular among beginners and professionals alike. It possesses a rich ecosystem of libraries that support data science, machine learning, artificial intelligence, image processing, and web development. Libraries such as NumPy, Pandas, TensorFlow, and OpenCV have contributed significantly to Python's dominance in modern research and technology development.

CNN

Convolutional Neural Network (CNN) is a specialized artificial neural network architecture designed for processing visual data such as images and videos. CNN extracts meaningful features from image inputs through convolutional, pooling, and activation layers, which are subsequently used for classification, segmentation, or object detection tasks.

In practical applications, CNNs are commonly implemented using Python and deep learning frameworks such as TensorFlow, PyTorch, and Keras. These frameworks provide efficient tools for building, training, and evaluating CNN models. CNNs are particularly effective in computer vision applications because they can automatically learn hierarchical feature representations from image data.

According to Goodfellow et al. (2016), CNNs significantly reduce the need for manually engineered features because they can automatically learn relevant feature representations directly from raw data during training.

TensorFlow

TensorFlow is an open-source software library developed by Google for numerical computation and machine learning, particularly deep learning. In CNN development, TensorFlow provides an efficient and flexible programming interface for building neural network architectures, training models, and optimizing parameters through backpropagation and gradient-based optimization techniques.

TensorFlow utilizes tensor data structures, which are multidimensional arrays particularly suitable for processing digital images. Features such as automatic differentiation, GPU acceleration, and support for complex neural network architectures have made TensorFlow one of the primary tools for developing computer vision applications, including face detection, image classification, and object tracking.

According to Abadi et al. (2016), TensorFlow was designed to provide high performance and portability, enabling machine learning models to be deployed across desktops, servers, and mobile devices.

OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source software library designed to support computer vision and machine learning applications. OpenCV provides numerous functions and modules that allow computers to capture, process, and analyze visual data automatically, including face detection, object recognition, feature extraction, and motion tracking.

OpenCV is widely used in computer vision because it can efficiently interact with cameras and various image and video formats. It supports real-time image processing and is compatible with programming languages such as Python and C++, as well as operating systems including Windows, macOS, and Linux.

MobileNetV2

MobileNetV2 is a Convolutional Neural Network architecture specifically designed for resource-constrained environments such as mobile devices and embedded systems. MobileNetV2 improves upon its predecessor, MobileNetV1, by providing enhanced efficiency and accuracy while maintaining a lightweight computational footprint.

In practice, MobileNetV2 is commonly employed as a feature extractor in CNN pipelines and as a backbone network for lightweight object detection frameworks, including TensorFlow Lite and TensorFlow Mobile.

Anaconda Navigator

Anaconda Navigator is a desktop-based graphical user interface included in the Anaconda distribution that simplifies the management of development environments, libraries, and applications for data science, machine learning, and scientific computing.

Navigator enables users to launch applications such as Jupyter Notebook, JupyterLab, Spyder, and Visual Studio Code without relying on command-line operations. It also includes an integrated package manager that facilitates the installation, updating, and removal of Python and R packages.

a) JupyterLab

JupyterLab allows users to create, edit, execute, and organize notebooks, Python scripts, shell terminals, Markdown documents, CSV files, and data visualizations within a unified workspace.

b) Jupyter Notebook

According to Kluyver et al. (2016), Jupyter Notebook is part of Project Jupyter, which supports multiple programming languages through a kernel architecture, with Python being the most widely used language in data science and machine learning.

One of Jupyter Notebook's key features is its cell-based structure, allowing users to combine executable code with explanatory text. It also supports exporting notebooks into multiple formats such as HTML, PDF, and Python scripts, while integrating seamlessly with libraries including NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow (Kluyver et al., 2016).

c) Conda Package Manager

Conda Package Manager is a package and environment management tool included with the Anaconda distribution. Conda enables users to efficiently install, update, remove, and manage software libraries and dependencies for Python, R, and other supported programming languages.

3. Materials and Method

Research Data

This study employed a quantitative approach using a computational experimental method to detect students' learning concentration levels through visual behavior analysis using the Convolutional Neural Network (CNN) algorithm. The data used were obtained from a public dataset containing images of students' behaviors and facial expressions during classroom learning activities. The dataset was selected because it provides variations in poses, expressions, and student activities that represent real classroom learning conditions.

The data were subsequently reclassified based on behaviors considered to reflect students' learning concentration levels. The behavioral categories used in this study included sleeping (sleeping in class), turning around (looking backward or sideways), looking forward (looking forward), reading, and writing. These five categories were selected because they represent both focused and unfocused conditions during learning activities.

The categorized dataset was then divided into 90% training data and 10% testing data using a randomized split method to maintain a proportional distribution of data across each class. Before being used in the model training process, all images underwent preprocessing stages, including image resizing, pixel value normalization, and data augmentation through rotation and flipping to increase data diversity and reduce the risk of overfitting.

System Development

The system development process consisted of several stages, including data collection, preprocessing, CNN model construction, model training, and real-time implementation.

During the data collection stage, a public dataset containing images of students' faces and behaviors was collected and subsequently reclassified into predefined behavioral categories. The next stage involved data preprocessing aimed at improving the quality of the input data. This process included face detection using Multi-task Cascaded Convolutional Networks (MTCNN), face region cropping, image resizing, pixel normalization to a range of 0–1, and data augmentation through rotation, lighting adjustments, and flipping.

The model used in this study was a Convolutional Neural Network (CNN) consisting of several main layers, namely a convolutional layer, Rectified Linear Unit (ReLU), max pooling layer, fully connected layer, and softmax output layer. The convolutional layer was used to extract spatial features from facial images, while the pooling layer functioned to reduce feature dimensions without losing important information. The extracted features were then processed through a fully connected layer to generate classifications of students' concentration levels.

The model training process was conducted using the previously processed training data. Model optimization was performed using the Adam algorithm with a categorical cross-entropy loss function. The training parameters included batch size, learning rate, and the number of epochs, which were adjusted to obtain optimal model performance.

After the training process was completed, the model was implemented in a real-time system using a webcam as the video input source. Each video frame was processed through face detection, preprocessing, and classification stages using the trained CNN model. The classification results were then displayed directly to indicate students' concentration levels based on the detected behaviors.

System Testing and Evaluation

Testing was conducted to evaluate the performance of the CNN model and the developed concentration detection system. The evaluation included classification accuracy testing, face detection capability testing, system robustness testing against environmental variations, and real-time implementation performance testing.

Accuracy testing was performed using testing data that were not involved in the training process. Model performance was evaluated using accuracy, precision, recall, and F1-score metrics. In addition, a confusion matrix was used to analyze the distribution of prediction results across each behavioral category.

The face detection capability was tested using a webcam to ensure that the system could simultaneously detect and track students' faces. The evaluation was carried out by comparing detection results with the actual conditions (ground truth) and observing the consistency of face tracking across video frames.

System robustness testing was conducted under various environmental conditions, including different lighting conditions and facial orientations. The purpose of this testing was to determine the model's robustness when faced with changes in real classroom environments.

Real-time performance evaluation was conducted by running the system directly using a webcam. The observed parameters included the number of frames processed per second (Frames Per Second/FPS) and the system's responsiveness in displaying classification results in real time. In addition, user experience evaluation was conducted through interviews with teachers or observers to assess the ease of use and implementation potential of the system in learning environments.

Research Workflow

The study began with the collection and labeling of student behavioral data. Subsequently, data preprocessing was carried out, including face detection, image size normalization, and data augmentation. The processed data were then used to train the CNN model and validated using testing data. After the model achieved the expected performance, offline testing was conducted using classification evaluation metrics, followed by testing under various environmental conditions and real-time implementation using a webcam. The final stage consisted of evaluating the testing results and analyzing system performance to determine the effectiveness of the model in detecting students' learning concentration levels in the classroom.

4. Results and Discussion

Implementation and Testing

System Workflow

a. Dataset Splitting

The dataset used in this study is a public dataset uploaded by phamluhuynhmai on Kaggle entitled "Classroom Student Behaviors." The dataset consists of 252,000 images classified into seven categories of student behavior in the classroom, namely: 1) Looking forward, 2) Raising hand, 3) Reading, 4) Sleeping, 5) Standing, 6) Turning around, 7) Writing

The relabeling of the "Classroom Student Behaviors" dataset into four learning concentration categories was carried out based on the operational definitions established in Chapter II. In this study, learning concentration was measured through observable student behaviors in the classroom using the Behavior Observation of Students in Schools (BOSS) instrument.

The mapping of behavioral categories to concentration levels is as follows:

Table 1. Theoretical Foundation for Mapping Learning Behaviors into Learning Concentration Levels.

Concentration Level	Behavior	Theoretical Basis
Highly Focused	Raising hand	Active Engagement (BOSS) [6]; Chin & Osborne [9]
Focused	Looking forward, Reading, Writing	Active/Passive Engagement (BOSS) [6]; Guthrie & Wigfield [10]
Less Focused	Turning around, Standing	Off-task Motor (BOSS) [6]; Finn & Zimmer [11]
Unfocused	Sleeping	Off-task Passive (BOSS) [6,8]

To accelerate the training process, the author reduced the dataset size to 36,765 images, consisting of:

a) Highly Focused: represented by the behavior raising hand, totaling 4,535 images.
b) Focused: represented by the behaviors looking forward, reading, and writing, totaling 14,306 images.
c) Less Focused: represented by the behaviors standing and looking around (turning around), totaling 9,439 images.
d) Unfocused: represented by the behavior sleeping, totaling 4,761 images.

Furthermore, the data were divided into training and testing sets with a ratio of 90:10.

a) Training data (90%): 4,030 highly focused images, 12,716 focused images, 8,434 less focused images, and 4,231 unfocused images.
b) Testing data (10%): 505 highly focused images, 1,590 focused images, 1,005 less focused images, and 530 unfocused images.

b. Data Preprocessing

The data preprocessing stage was conducted to ensure that all images had a consistent format and size before being fed into the model. All images were resized to 224 × 224 pixels, according to the standard input size of the MobileNetV2 architecture.

Normalization was performed using the preprocess_input function provided by the MobileNetV2 library to adjust pixel values according to the standards used when MobileNetV2 was trained on the ImageNet dataset. This ensured that the pixel value distribution of the new data aligned with the pre-trained weights utilized by the model.

In addition, data augmentation was applied to improve the model's generalization capability under various real-world conditions. The augmentation techniques included image rotation up to 15 degrees, zooming up to 10%, and horizontal flipping. This augmentation was applied to the training and validation datasets but not to the testing dataset to ensure objective evaluation results.

The training and validation data split was performed automatically using the parameter validation_split = 0.1, meaning that 90% of the data were used for training and 10% for validation.

All information regarding class-to-index mapping was stored in the classes4.json file. This storage is important to ensure that the inference process in later stages refers to consistent class labels.

c. Model Architecture

The model used in this study was based on MobileNetV2 with pre-trained weights obtained from training on the ImageNet dataset. MobileNetV2 was selected due to its computational efficiency, relatively small model size, and ability to extract complex visual features without significantly compromising accuracy.

In the initial stage, all MobileNetV2 weights were frozen so that they were not updated during the training process. This strategy aimed to preserve the general feature representation capabilities learned during pre-training.

The classifier head of the model was specifically designed for this study. Following the MobileNetV2 convolutional layers, a Global Average Pooling layer was employed to reduce the feature dimensions into a more compact one-dimensional vector.

Subsequently, a Dense layer with 128 neurons was added to connect the extracted features to the classification process and was activated using the ReLU activation function. To prevent overfitting, a Dropout layer with a rate of 0.3 was applied.

Finally, the output layer consisted of four neurons representing the four student concentration categories (highly focused, focused, less focused, and unfocused) and was optimized using the categorical cross-entropy loss function.

d. Model Training Process

The model was trained using the Adam optimization algorithm with a learning rate of 0.0002. This parameter was selected to provide stable weight updates and prevent oscillation during the learning process.

The categorical cross-entropy method was used as the loss function because it is suitable for multi-class classification problems.

To maintain model performance, two callback mechanisms were employed:

a) EarlyStopping, which terminates training if the validation accuracy does not improve for three consecutive epochs while restoring the best model weights obtained.
b) ModelCheckpoint, which saves the best-performing model based on the highest validation accuracy.

The training process was conducted for 10 epochs using a data generator that supplied data incrementally in batches of 16 samples. During training, each epoch was followed by validation using the validation dataset to monitor model performance.

e. Model Evaluation and Storage

After training was completed, the model was evaluated using the testing dataset, which comprised 10% of the total dataset. This evaluation provided an objective assessment of the model's ability to classify previously unseen data.

The test accuracy value was recorded as the final performance benchmark of the model.

The trained model was then saved in the .keras format to facilitate its use in future implementation stages or subsequent research. In addition to the final model, the best-performing model generated during training was automatically saved through the ModelCheckpoint callback.

CNN Classification Accuracy Testing Results

a. CNN Classification Accuracy Testing

Accuracy testing was conducted by dividing the dataset into training and testing sets. The MobileNetV2-based CNN model was evaluated using the metrics of accuracy, precision, recall, and F1-score.

The accuracy formula used is:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of test data}} \times 100\%$$

The testing results on the testing dataset (5,881 images) are presented as follows.

Table 2. Dataset Testing Results.

Class	Precision	Recall	F1-Score	Support
Focused	0.9859	0.6032	0.7485	2,543
Less Focused	0.8448	0.7782	0.8101	1,686
Highly Focused	0.5406	0.9739	0.6953	806
Unfocused	0.6288	0.9811	0.7664	846
Overall Accuracy	-	-	0.7585	5,881
Macro Average	0.7500	0.8341	0.7551	5,881
Weighted Average	0.8330	0.7585	0.7614	5,881

Table 2 shows the classification report of the CNN model on the testing dataset. The overall accuracy reached 75.85%, with the highest precision achieved by the Focused class (0.9859), while the highest recall values were observed in the Highly Focused (0.9739) and Unfocused (0.9811) classes.

These results indicate that the model can recognize extreme categories effectively; however, the precision remains relatively low, causing some misclassifications into these categories.

Figure 2. Confusion Matrix.

Figure 2 illustrates the confusion matrix generated from the testing dataset. The rows represent the true labels, while the columns represent the model predictions.

a) The Focused class was recognized relatively well, with 1,534 correct predictions. However, 664 cases were incorrectly classified as Highly Focused, and 224 cases were misclassified as Less Focused. This indicates visual similarities between focused students and other categories.
b) The Less Focused class achieved 1,312 correct predictions, although 367 cases were incorrectly classified as Unfocused. Nevertheless, this class remained relatively stable, with balanced precision and recall values.
c) The Highly Focused class was recognized with 785 correct predictions, but exhibited low precision because many incorrect predictions originated from the Focused class (664 cases classified as Highly Focused). Consequently, the class achieved high recall but low precision.
d) The Unfocused class demonstrated relatively consistent performance with 830 correct predictions, although its precision remained limited because approximately 316 samples from other classes were incorrectly assigned to this category.

b. Face Detection Capability Testing

Face detection testing was conducted using a webcam to detect student faces directly in the classroom. The test employed a 720p webcam to recognize and track student behaviors and facial expressions in real time within a single frame. Evaluation was carried out based on two primary aspects. The testing results showed that the system was able to maintain face IDs with high consistency when students were positioned close to the camera (approximately 1 meter, estimated from desk and chair placement) and under sufficiently bright lighting conditions. However, tracking performance declined when rapid movements occurred, the classroom became crowded, or the background contained multiple colors.

c. System Robustness Testing Under Environmental Variations

Environmental variation testing was conducted to assess system robustness under different conditions, particularly lighting and facial orientation variations.

The testing included two primary scenarios:

a) Frontal face position: the student's face directly faces the computer webcam.
b) Side face position: the student's face is turned sideways or positioned at a significant angle.

The results demonstrated that the system could only detect faces effectively when the face was positioned directly toward the camera. When faces were viewed from the side or captured from extreme angles, the system failed to perform detection and tracking.

Table 3. Webcam Detection Results from Various Viewing Angles.

Face Position	Detection Accuracy	Tracking Consistency	Testing Notes
Frontal	>95%	High (stable)	Fast detection, consistent face IDs, functions well under sufficient lighting
Sideways / Tilted	<35%	Low (frequently lost)	Detection frequently fails, tracking becomes unstable, and accuracy decreases drastically

Based on these results, it can be concluded that facial orientation is a critical factor affecting system performance, whereas lighting variation has a relatively minor impact as long as illumination remains adequate.

d. Real-Time Performance Testing

Real-time performance testing was conducted to evaluate the extent to which the system could provide direct detection and classification in a classroom environment.

Three parameters were measured:

1) Frame per Second (FPS)

FPS was measured to evaluate the smoothness of real-time detection performance. FPS was calculated every 10 frames rather than every frame to obtain more stable measurements.

The FPS value was displayed directly in the output window using the cv2.putText function. FPS measurements were recorded over 20 seconds of classroom testing, and average values were documented.

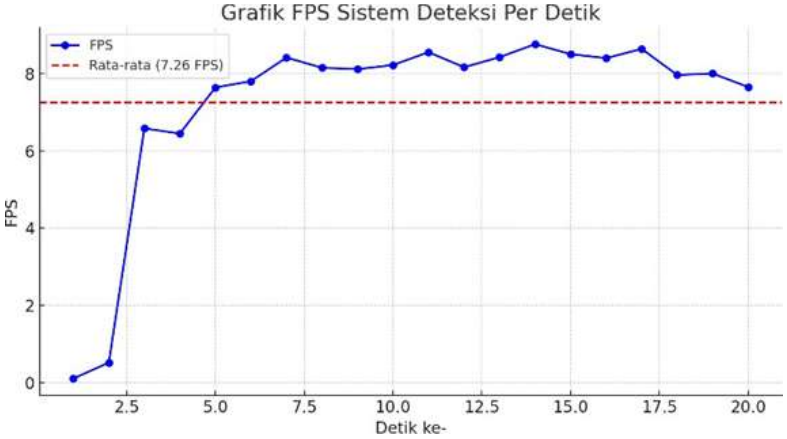

Figure 3. FPS Graph.

Based on Figure 3:

a) FPS was very low during the first and second seconds due to system initialization.
b) Starting from the third second, FPS increased and stabilized within the range of 6–9 FPS.

c) The red dashed line indicates an average FPS of 7.257, showing that the system was capable of processing approximately 7 frames per second.

2) User Experience (UX)

The subjective evaluation of the student learning concentration detection system based on facial expression images using CNN was obtained through questionnaires distributed to two respondent groups: a) Students (12 respondents). b) Teachers (8 respondents) at Skill Village Islamic School. The questionnaire was designed to evaluate detection accuracy, result clarity, user satisfaction, and the usefulness of the system in supporting the learning process.

a) Student Evaluation

Figure 4. Detection Result Suitability Diagram.

Kuisioner Kesesuaian Hasil Deteksi Konsentrasi Melalui Perilaku Belajar Siswa di SMK Skill Village Islamic School

Nama Siswa	Kejelasan Hasil Deteksi	Fokus (Ya/Tidak)	Aktivitas Saat Fokus	Kurang Fokus (Ya/Tidak)	Aktivitas Saat Kurang Fokus
Muhammad Jauhar	Cukup	Ya	Melihat ke depan	Tidak	Melihat ke depan
Fachri Syabil Al awwaz	Tidak Sesuai	Tidak	Melihat ke depan	Ya	Melihat ke depan
Syahla Kayyisyah	Sesuai	Ya	Melihat ke depan	Tidak	Melihat ke depan
Aulia Putri Nurron Qolbi	Cukup	Ya	Melihat ke depan	Tidak	Melihat ke depan
Safaraz Aufa	Sesuai	Tidak	Tertidur	Tidak	Tertidur
Zyekh Abdul	Cukup	Ya	Melihat ke depan	Tidak	Melihat ke depan
Nawwaf	Sangat Sesuai	Ya	Menulis	Tidak	Menulis
Almuntazar Mubarak	Cukup	Tidak	Lainnya : tertawa	Ya	Lainnya : tertawa
Ruhdiansah	Sesuai	Tidak	Lainnya : tertawa	Ya	Lainnya : tertawa
Rakha Nugraha	Cukup	Tidak	Lainnya : tertawa	Ya	Lainnya : tertawa
Ekalaya Klana	Sesuai	Ya	Membaca	Tidak	Membaca
Andini	Sesuai	Tidak	Menoleh	Ya	Menoleh

Tidak Fokus (Ya/Tidak)	Aktivitas Saat Tidak Fokus	Kepuasan (1-5)	Apakah Membantu (Ya/Tidak)
Tidak	Melihat ke depan	5	Ya
Tidak	Melihat ke depan	2	Tidak
Tidak	Melihat ke depan	4	Ya
Tidak	Melihat ke depan	3	Tidak
Ya	Tertidur	5	Ya
Tidak	Melihat ke depan	3	Ya
Tidak	Menulis	4	Ya
Tidak	Lainnya : tertawa	2	Tidak
Tidak	Lainnya : tertawa	4	Ya
Tidak	Lainnya : tertawa	5	Tidak
Tidak	Membaca	3	Ya
Tidak	Menoleh	4	Ya

Figure 5. Student Questionnaire Results.

Based on responses from 12 students, the majority stated that the system provided results that matched their classroom conditions. A total of 50% of students rated the detection results as appropriate or highly appropriate, while the remaining 50% rated them as moderately appropriate to inappropriate.

Regarding satisfaction, the average score was 3.67 out of 5, with 58% of students reporting satisfaction or high satisfaction. Additionally, 67% stated that the system helped them understand their learning concentration levels in class, while 33% felt it was not significantly helpful.

b) Teacher Evaluation

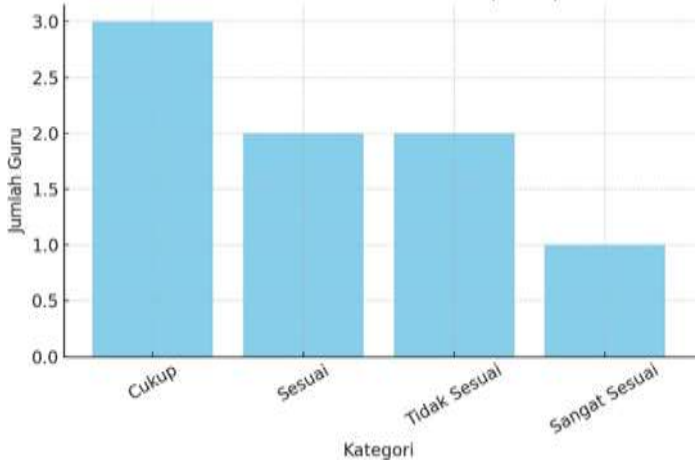

Figure 6. Suitability of Detection Results with Teacher Observations.

Kuisioner Kesesuaian Hasil Deteksi Konsentrasi Melalui Perilaku Belajar Siswa di SMK Skill Village Islamic School oleh Guru

Guru	Kesesuaian Hasil Deteksi	Membantu Memahami Kondisi Kelas	Manfaat Bagi Pembelajaran	Bersedia Menggunakan Kembali
Badrana Nabila	Sesuai	Membantu	Bermanfaat	Ya
Nadia Ulfa	Cukup	Cukup Membantu	Cukup Bermanfaat	Ya
Khurul Aina Alika	Sangat Sesuai	Membantu	Bermanfaat	Ya
Nazzahra Anggelina	Cukup	Cukup Membantu	Cukup Bermanfaat	Ya
Win Prayoga	Sesuai	Membantu	Bermanfaat	Ya
Jundi Kariman Husni	Cukup	Cukup Membantu	Cukup Bermanfaat	Ya
Widi Dwi	Tidak Sesuai	Tidak Membantu	Kurang Bermanfaat	Tidak
Ziyad Fernanda S	Tidak Sesuai	Tidak Membantu	Kurang Bermanfaat	Tidak

Figure 7. Teacher Questionnaire Summary

Teacher responses to the system were relatively diverse. Of the eight teachers surveyed: 3 teachers rated the detection results as appropriate or highly appropriate. 3 teachers rated them as moderately appropriate. 2 teachers rated them as inappropriate. Therefore, the suitability of the detection results can be categorized as moderate. Regarding informational usefulness: 3 teachers stated that the system was helpful. 3 teachers considered it moderately helpful. 2 teachers considered it unhelpful. Regarding benefits for learning: 3 teachers considered the system beneficial. 3 considered it moderately beneficial. 2 considered it less beneficial. Although the perceived benefits were not uniformly distributed, the majority of teachers maintained a positive attitude toward the system. Teacher willingness to reuse the system was also relatively high. A total of 6 teachers (75%) expressed willingness to use the system again, while 2 teachers (25%) indicated otherwise. This suggests a strong level of acceptance despite some reservations.

Final Testing Results

Based on the entire implementation and testing process, the CNN-based student learning concentration detection system demonstrated an overall testing accuracy of 75.85%. The highest precision value was achieved by the Focused category (0.9859), while the highest recall values were obtained by the Highly Focused (0.9739) and Unfocused (0.9811) categories. These findings indicate that the system is capable of recognizing extreme categories relatively well, although precision remains limited, resulting in some misclassifications.

Face detection testing showed that the system could consistently detect faces when students faced the camera directly under adequate lighting conditions. However, performance decreased significantly when facial angles exceeded 45° or when rapid movements occurred. These results indicate that facial orientation strongly affects system performance, whereas lighting variations have a limited effect provided that illumination remains sufficient.

From a real-time performance perspective, the system achieved an average processing speed of 7.25 FPS, which is adequate for classroom monitoring purposes, although it does not meet the standard threshold for smooth video processing (>15 FPS). This performance suggests hardware limitations, but the system remains functionally usable in educational environments.

Subjective evaluations from both students and teachers also indicated positive acceptance. Among the 12 students surveyed, 50% stated that the detection results were appropriate or highly appropriate. The Focused category appeared most frequently and was considered representative of behaviors such as reading, writing, and looking forward. Within the Unfocused category, sleeping behavior proved to be the most accurately and consistently

detected behavior. Student satisfaction reached an average score of 3.67 out of 5, with 58% expressing satisfaction and 67% reporting that the system helped them better understand their learning concentration.

Among the eight teacher respondents, perceptions varied. A total of 37.5% rated the system as moderately appropriate, 25% as appropriate, 12.5% as highly appropriate, and 25% as inappropriate. Regarding informational usefulness, 37.5% found it helpful, 37.5% considered it moderately helpful, and 25% found it unhelpful. Concerning educational benefits, 37.5% considered it beneficial, 37.5% moderately beneficial, and 25% less beneficial. Despite the varied opinions, teacher acceptance remained high, with 75% expressing willingness to use the system again in future classroom activities.

Overall, the testing results demonstrate that the CNN-based concentration detection system provides reasonably good classification performance, can operate in real time on standard hardware, and is positively received by both students and teachers. Nevertheless, further improvements are needed, particularly in enhancing classification accuracy for specific behaviors, improving detection stability under extreme facial angles, and increasing the practical benefits of the system in classroom learning environments.

5. Conclusion and Suggestion

Conclusion

The student learning concentration detection system was designed using the MobileNetV2-based Convolutional Neural Network (CNN) architecture. The development process involved several stages, including image preprocessing (resizing, normalization, and augmentation), model training, and camera integration. The system was also equipped with a dashboard that displays detection results in real time.

The classification of student behaviors into learning concentration levels was performed by mapping seven behaviors from the public Classroom Student Behaviors dataset into four categories. The behavior raising hand was categorized as highly focused; looking forward, reading, and writing were categorized as focused; standing and turning around were categorized as less focused; and sleeping was categorized as unfocused. Thus, student behaviors could be translated into concentration levels based on the operational definitions employed in this study.

The testing results showed an overall system accuracy of 75.85%, with the highest precision achieved in the focused class (0.9859), while the highest recall values were obtained in the highly focused (0.9739) and unfocused (0.9811) classes. Confusion matrix analysis revealed that the focused class was detected most consistently, whereas the highly focused and unfocused classes were still frequently misclassified as focused. In real-time testing, the system achieved an average processing speed of 7.25 FPS, which was sufficiently stable for monitoring purposes, although it did not yet meet the standard requirements for smooth video performance.

Subjective evaluation through questionnaires distributed to students and teachers indicated positive acceptance of the system. Among the 12 participating students, 50% considered the detection results to be appropriate or highly appropriate, with an average satisfaction score of 3.67 out of 5, while 67% believed that the system helped them better understand their learning concentration. Among the eight participating teachers, some considered the system to accurately reflect classroom conditions, others rated it as moderately appropriate, and a small proportion considered it unsuitable. Regarding usefulness, most teachers evaluated the system as moderately useful to useful, and 75% expressed their willingness to use the system again in the future. These findings suggest that the system has significant potential for implementation in educational settings, although further improvements are required to enhance detection accuracy and maximize its practical benefits.

Suggestion

Based on the findings of this study, several aspects should be considered for future development. First, dataset balancing is necessary to achieve more consistent performance across all classes, particularly in the highly focused and unfocused categories, which still exhibited relatively low precision due to frequent misclassification as the focused class. Second, the system should be enhanced with a more robust face detection algorithm capable of handling extreme facial angles, rapid movements, and crowded classroom environments.

Furthermore, the real-time testing results, which averaged only 7.25 FPS, indicate the need for hardware optimization. The use of higher-resolution cameras and devices equipped with more powerful GPUs is expected to improve processing speed and enable smoother operation in real classroom settings. Additional experiments should also be conducted in classrooms with larger numbers of students and more diverse learning conditions to evaluate the system's performance on a broader scale.

From an implementation perspective, integrating the system with an interactive learning dashboard that presents periodic visualizations of students' concentration levels could provide valuable support for teachers. Such visual analytics would enable educators to better understand concentration patterns and implement appropriate interventions during the learning process.

References

Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI)* (pp. 265–283). USENIX Association.

Alperin, A., et al. (2023). Convergent validity of the Behavior Observation of Students in Schools (BOSS) form. *Psychology in the Schools, 60*(10), 2031–2045.

Alruwais, N., & Zakariah, M. (2025). Detecting student engagement with convolution neural network and facial expression recognition. *Technical Sciences Journal, 42*(2), 943–961. https://doi.org/10.18280/ts.420229

Ansari, M. F., Kasprowski, P., & Obetkal, M. (2021). Gaze tracking using an unmodified web camera and convolutional neural network. *Applied Sciences, 11*(19), 9068. https://doi.org/10.3390/app11199068

Arifin, S., Aisjah, A. S., Fatima, A. N., & Mahmudah, H. (2020). Design and development of a system for monitoring student attention and concentration using CNN model and face landmark detection. In *Proceedings of the 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 170–175). IEEE. https://doi.org/10.1109/ISRITI51436.2020.9315513

Avon-Washington County Schools. (2015). *Systematic behavior observation form.* Avon Central School District.

Dewan, M. L., Sharma, R., & Kumar, M. (2019). Engagement detection in online learning: A review. *Smart Learning Environments, 6*(1), 1–21. https://doi.org/10.1186/s40561-019-0094-0

Dimyati, & Mudjiono. (2009). *Belajar dan pembelajaran*. Rineka Cipta.

Gonzalez, R. C., & Woods, R. E. (2008). *Digital image processing* (3rd ed.). Pearson Prentice Hall.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gupta, A., D'Mello, S., & Baker, R. (2017). Data set for affective states in e-learning environments (DAiSEE). In *Proceedings of the 8th International Conference on Affective Computing and Intelligent Interaction* (pp. 236–242). IEEE.

Hintze, J. M., Volpe, R. J., & Shapiro, E. S. (2002). Best practices in the systematic direct observation of student behavior. In A. Thomas & J. Grimes (Eds.), *Best practices in school psychology IV* (pp. 999–1020). National Association of School Psychologists.

Jia, Q., & He, J. (2024). Student behavior recognition in classroom based on deep learning. *Applied Sciences, 14*(17), 7981. https://doi.org/10.3390/app14177981

Kluyver, T., et al. (2016). Jupyter notebooks—A publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press.

Li, X., Song, D., & Lu, B.-L. (2016). Emotion recognition based on EEG using hybrid deep learning model. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 1013–1018). IEEE.

Qi, J., Zhang, H., Liu, X., Yang, W., & Zhang, M. (2024). Application of face detection for learning engagement in the classroom. *Electronics, 13*, 149. https://doi.org/10.3390/electronics13010149

Qi, Y., Zhuang, L., Chen, H., Han, X., & Liang, A. (2023). Evaluation of students' learning engagement in online classes based on multimodal vision perspective. *Electronics, 13*(1), Article 149. https://doi.org/10.3390/electronics13010149

Rasiban, J., & Praja Raymond Maruli, S. (2022). Penerapan data mining untuk memprediksi penerimaan peserta didik baru. *Journal of Military Science and Technology, 3*, 22–29. https://doi.org/10.54930/1859-1043.j.mst.83.2022.22-29

Slameto. (2010). *Belajar dan faktor-faktor yang mempengaruhinya*. Rineka Cipta.

Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2nd ed.). Springer.

Wang, Z., Wang, M., Zeng, C., & Li, L. (2024). *Multi-scale deformable transformers for student learning behavior detection in smart classroom* (arXiv:2410.07834). arXiv. https://arxiv.org/abs/2410.07834

Whitehill, J., Serpell, Z., Lin, Y.-C., Foster, A., & Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing, 5*(1), 86–98. https://doi.org/10.1109/TAFFC.2014.2316244

Winkel, W. S. (2009). *Psikologi pengajaran*. Gramedia.

Zheng, W.-L., & Lu, B.-L. (2015). Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development, 7*(3), 162–175.

Zhou, H., Jiang, F., Si, J., Xiong, L., & Lu, H. (2023). *StuArt: Individualized classroom observation of students with automatic behavior recognition and tracking* (arXiv:2111.03127v3). arXiv. https://arxiv.org/abs/2111.03127