International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 5
ISSN: 2088-5334

Facial and Body Gesture Recognition for Determining Student Concentration Level

Xian Yang Chan [a], Tee Connie [a,*], Michael Kah Ong Goh [a]

[a] *Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450, Melaka, Malaysia*
Corresponding author: [*]tee.connie@mmu.edu.my

***Abstract*— Online learning has gained immense popularity, especially since the COVID-19 pandemic. However, it has also brought its own set of challenges. One of the critical challenges in online learning is the ability to evaluate students' concentration levels during virtual classes. Unlike traditional brick-and-mortar classrooms, teachers do not have the advantage of observing students' body language and facial expressions to determine whether they are paying attention. To address this challenge, this study proposes utilizing facial and body gestures to evaluate students' concentration levels. Common gestures such as yawning, playing with fingers or objects, and looking away from the screen indicate a lack of focus. A dataset containing images of students performing various actions and gestures representing different concentration levels is collected. We propose an enhanced model based on a vision transformer (RViT) to classify the concentration levels. This model incorporates a majority voting feature to maintain real-time prediction accuracy. This feature classifies multiple frames, and the final prediction is based on the majority class. The proposed method yields a promising 92% accuracy while maintaining efficient computational performance. The system provides an unbiased measure for assessing students' concentration levels, which can be useful in educational settings to improve learning outcomes. It enables educators to foster a more engaging and productive virtual classroom environment.**

***Keywords*— Vision transformer; random projection; facial expression recognition; gesture recognition; concentration level prediction.**

Manuscript received 25 Nov. 2022; revised 17 Mar. 2023; accepted 20 Jun. 2023. Date of publication 31 Oct. 2023.

I. INTRODUCTION

The rise of online education has led to a shift from physical classrooms to virtual learning environments [1], [2]. While online classes offer convenience and accessibility, they present challenges in maintaining student concentration. Unlike in traditional classrooms, where teachers can gauge student's focus through facial expression and body language, virtual settings lack the same level of supervision and interaction [3]. In light of the COVID-19 pandemic, the transition to online learning has been necessary but not without flaws [4], [5]. Many students struggle to stay engaged in virtual classes due to the relaxed environment, distractions, and the absence of constant monitoring by teachers [6]. This lack of attention is particularly problematic for students with low concentration, as it hinders their ability to retain knowledge [7].

To address this issue, a concentration tracker becomes crucial in helping teachers understand their students in virtual classrooms. Computer vision and machine learning techniques [8], [9] can be employed to identify signs of restlessness and loss of focus. In a physical classroom, body posture, facial expressions, and emotions are effective indicators of concentration, but replicating this in a virtual setting is challenging. However, computer vision can provide a solution by analyzing body traits such as head tilt angle and spinal erectness, which correlate with mental attentiveness [10]. Facial expression can also reveal emotions and feelings, helping to determine a student's level of engagement. Additionally, signs of drowsiness, evident through facial expressions and gaze, can serve as reliable indicators of attention during lectures [11].

While online education offers advantages in terms of accessibility and flexibility, ensuring student concentration and engagement remains a significant concern. Implementing a concentration tracker utilizing computer vision and machine learning can aid teachers in understanding and addressing the attentiveness of their students in virtual classrooms. By analyzing body language, facial expressions, and signs of drowsiness, this technology can provide valuable insights into student engagement and help improve the overall learning experience in the online education landscape.

Towards this end, a deep learning approach coined as **R**andomized **Vi**sion **T**ransformer (RViT) is presented in this paper. The proposed method that relies on the vision transformer model offers an efficient way to capture both global and local dependencies on the data samples to capture characteristics of the students' learning behavior. Moreover, the incorporation of random projection techniques contributes to reducing the computational demands of the system, making it more efficient and scalable. RViT is particularly well-suited for processing large-scale visual data, demonstrating remarkable performance in handling diverse and extensive datasets of concentration level analysis. Additionally, using random projection further enhances RViT's generalization capabilities, enabling it to effectively classify and recognize patterns in unseen concentration-level samples.

Several studies have investigated different conventional approaches to estimating students' concentration levels. KNN and DT algorithms were implemented for concentration detection. Kinect One sensor was utilized to gather 2D and 3D information on facial and body attributes. However, the highest accuracy achieved by the method is only 75.3% [12]. In 2017, SVM and LR were proposed to estimate the concentration level of students based on their facial expressions, head stance, and eye gaze. Clustering of Static-Adaptive Correspondence for Deformable Object Tracking (CMT) and OpenFace toolkit's Facial Action Unit (AUs) were adopted to analyze individual faces and pre-processing. The best accuracy achieved by this method is 90% [13].

Several wearable devices, including smart glass, smart cap, and smart pen, were used to collect data on students in the classroom to analyze their concentration levels. The data were then trained using J48 DT, RF, and SVM. Of the three algorithms, J48 DT achieved the highest accuracy of 82% based on the performance of the concentration inference engine [14].

A framework was developed based on eye states, utilizing Gabor wavelets combined with various classifiers, including KNN, Naïve Bayes (NB), and SVM. Principal Component Analysis (PCA) was also proposed to analyze complex data and extract relevant features. The best performance achieved by the methods was the combination of Gabor with the SVM algorithm, with an accuracy of 93.1% [15]. Last but not least, Tobii 4c Eyetrackers device was employed in collecting the eye-gaze data of individuals. Several algorithms, such as SVM, KNN, and Extreme Gradient Boosting (XBG) were employed for data training. With this method, XGB has the highest accuracy of 77%, followed by SVM with 73% accuracy [16].

In contrast to conventional methods, deep learning models can conduct expression classification tasks in an end-to-end manner, which often produces more accurate results. Some of the algorithms that were implemented in the past include You-Only-Look-Once (YOLO), Convolutional Neural Network (CNN), and Deep Neural Network (DNN). An autonomous agent that employed CNN and ResNet101 layers for face and concentration detection was developed. A Proportional Integral controller (PI) was employed in the agent camera to deal with underexposure and overexposure of the environment. This method achieved an accuracy of 85% [17].

In 2020, a Dempster-Shafer theory (DS-based) evaluation algorithm was proposed by measuring students' Euler angles of their facial attitude to determine the concentration level. The proposed method includes three modules: facial detection, attitude angle measurement, and concentration detection. Using CNN, this method achieved an accuracy of 85.3% [18]. On the other hand, the YOLO v3 algorithm is utilized to predict student behavior in the classroom. ImageAI was used to train the dataset to help the algorithm identify objects, faces, and eyes. Tesla GPUs and TensorFlow platform were also employed. The model achieved an accuracy of 88.6% [19].

A posture-based attentivity detection model using CNN and OpenPose was also created. The images dataset was tagged with five postures: 'attentive', 'head rested on hand', 'leaning back', 'writing', and 'not looking at the screen'. The model was trained for 20 epochs only and achieved 99.82% accuracy [3].

On the other hand, researchers captured video datasets of various participants ranging from 18-30 years old. The datasets were then pre-processed into images, and each image was classified based on three classes: Attentive, Partially Attentive, and Inattentive. In this experiment, CNN algorithm was employed and achieved an accuracy of 90.6%. Table I provides a concise overview of the methods discussed and their corresponding performance results [20].

TABLE I
A SUMMARY OF RELATED WORKS

Author	Method	Dataset	Accuracy (Highest)
Zaletelj *et al.* (2017) [12]	KNN, DT	Self-collected dataset	75.3%
Thomas *et al.* (2017) [13]	SVM, LR	Self-collected dataset	90%
Zhang *et al.* (2017) [14]	J48 DT, RF SVM	Self-collected dataset	82%
Deng *et al.* (2018) [15]	KNN, NB, SVM	CEW dataset	93.1%
Veliyath *et al.* (2019) [16]	SVM, KNN. XBG	Self-collected dataset	77%
Canedo *et al.* (2018) [17]	CNN	Self-collected dataset	85%
Li *et al.* (2020) [18]	CNN-Dempster-Shafer	Self-collected dataset	85.3%
Mindoro *et al.* (2020) [19]	YOLO v3	Self-collected dataset	88.6%
Revadekar *et al.* (2020) [3]	CNN	Self-collected dataset	99.82%
Shamika *et al.* (2021) [20]	CNN	DAISEE dataset	90.6%

II. MATERIAL AND METHOD

This section presents the details of the proposed RViT method. Instead of training the model directly, randomized projection is applied in the layers preceding the fully connected layer to transform the high-dimensional data into a lower-dimensional space, reducing training time and resource requirements. The model's predictions are used to classify the images extracted from video data into four different categories: normal concentration level, early stage of focus loss, mid-stage of focus loss, and late stage of focus loss. To enhance classification accuracy, majority voting is used to consider the collective probability of a set of images rather

than relying solely on the prediction of a single image. By considering multiple predictions, the overall accuracy of the classification process is improved. A block diagram of the proposed model is depicted in Fig. 1.

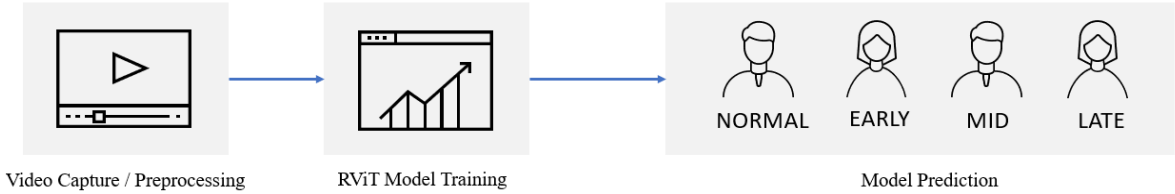

Fig. 1 Block diagram of the proposed method

A. Vision Transformer (ViT)

Vision Transformer (ViT) has emerged as a popular deep learning model for computer vision, drawing inspiration from the successful application of Transformers in natural language processing [21]–[23]. While Convolutional Neural Networks (CNNs) have been widely used for image classification, they have certain limitations, including limited global context modeling and difficulties in capturing long-range dependencies [24], [25]. ViT overcomes these drawbacks by adopting the Transformer architecture, originally designed for sequence transduction tasks. Instead of convolutions, ViT treats images as sequences of patches, flattened and transformed into lower-dimensional representations using linear projections. This approach allows ViT to capture relationships between different elements in the image and achieve impressive results in visual recognition tasks. The architecture of the ViT is discussed in the following sections.

1) Patch Embedding: Given an input image, $\epsilon \mathbb{R}^{H\times W\times C}$, where H represents the height of the image, W represents the width, and C represents the number of channels. The image pass through a series of transformations to produce a feature vector with dimensions (N+1, D), where N refers to the number of inputs or samples, and D signifies the size of the latent vector. This process can be described by [26].

$$Z_0 = \left[X_{class}; X_p^1 E; X_p^2 E; \ldots; X_p^N E\right] + E_{pos}, \quad E \in R^{(P^2.C)\times D}, E_{pos} \in R^{(N+1)\times D} \qquad (1)$$

where Z_0 is the resulting tensor obtained from MLP, X_{class} represents the class of the input image, and the vectors $X_p^1 E; X_p^2 E; \ldots; X_p^N E$ represents the additional input features. E is a matrix with dimensions $R^{(P^2.C)\times D}$, where R denotes the real number space, and P signifies the resolution of each image patch. $Epos \in R^{(N+1)\times D}$ represents the learned positional embedding matrix. It captures positional information and is added to the input.

Each input image is divided into patches with dimensions (P, P, C). These patches are subsequently flattened, creating N linear vectors with a shape of (1, P^2C). Each resulting patch is then multiplied by a trainable embedding tensor, enabling a linear projection of the patches into a D-dimensional space. This dimensional embedding is consistent throughout the architecture and is used in various components. The result is N embedded patches with a (1, D) shape.

In order to obtain an overall representation of the patches, a learnable token [cls] with a shape of (1, D) is introduced. [cls] token, inspired by BERT [27], summarizes the patch representations. Only the final representation associated with this token is utilized for downstream tasks, such as classification. A trainable positional embeddings tensor, denoted as Epos, is introduced to incorporate positional information. It has the same shape as the patch embeddings, allowing for the inclusion of details for each patch.

The resulting tensor, Z_0, serves as the initial input for the stacked transformer encoders. These encoders constitute the second component of the architecture. Each transformer encoder receives a tensor as input and generates an output tensor of the same dimension. Fig. 2 illustrates an image in the patch embedding layer.

Fig. 2 A patch embedding image

2) Transformer Encoder: This layer is responsible for processing and refining the patch embeddings generated by the Patch Embedding Layer. The network utilizes a stack L transformer encoder to learn more abstract features from the embedded patches. The transformer encoders contribute to extracting higher-level representations and enable the model to capture complex patterns and relationships within the image data. This process can be described by the given equations [26].

$$Z'_\ell = MSA\left(LN(Z_{\ell-1})\right) + Z_{\ell-1}, \quad \ell = 1\ldots L \qquad (2)$$

$$Z_\ell = MLP\left(LN(Z'_\ell)\right) + Z'_\ell, \quad \ell = 1\ldots L \qquad (3)$$

where Z_ℓ^i is the i-th token of the ℓ-th layer, MSA (·) is the Multi-Head-Self-Attention layer, MLP(·) is the Multi-Layer perceptron layer, and LN(·) is the layer normalization block.

The encoder component in the architecture consists of MSA mechanism and a two-layer MLP. Between these two

parts, layer normalization and residual connections are applied to stabilize hidden state dynamics and alleviate the vanishing gradient problem in deep architectures. During layer normalization, the hidden state is normalized by scaling it using the mean and standard deviation specific to each training sample. The state is scaled with each training sample's mean and standard deviation during layer normalization. This aids in the normalization of values and the efficiency of training. The normalized features undergo a multiplication operation with a learnable scaling factor and are subsequently added to a learnable shifting factor.

Residual connections play a crucial role in gradient propagation. These alternative paths for gradients allow for the flow of gradients, addressing the problem of vanishing gradients that can occur in deep architectures. By including residual connections, the gradients have multiple routes to flow, enabling more effective training. The trainable weights in the encoder component are located within the MSA mechanism and the MLP. The MLP has two weight matrices: W_h of shape (d, d_{mlp}) and W_o of shape (d_{mlp}, d), where d represents the input dimension and d_{mlp} is the dimension of the intermediate layer in the MLP. These weight matrices are learned during training, allowing the model to adapt and capture important features in the data.

3) Multi-Head Attention: The MSA is an integral part of each of the L stacked transformers. It involves a set of equations that are applied to process the input and capture important dependencies and relationships within the data [28].

$$[q, k, v] = zU_{qkv} \tag{4}$$

$$A = softmax(qk^T/\sqrt{D_h}) \tag{5}$$

$$SA(z) = Av \tag{6}$$

$$MSA(z) = [SA_1(z); SA_2(z); \ldots; SA_k(z)]U_{msa} \tag{7}$$

The previous encoder's hidden state is divided within the encoder component, yielding K feature tensors with a shape of (n, d_h). This approach enables it to capture and learn from distinct aspects of the representation. For every head, three matrices: Qi, Ki, and Vi are multiplied, each having a dimension of (d_h, d_h). This corresponds to Equation 5, where the matrix U has dimensions (d, $3d_h$), representing the three matrices for each head.

Qi, Ki, and Vi represent projections of the input into three subspaces. Qi can be seen as learned projections of the patch of interest, while Ki represents comparisons with other patches. Vi and Ki are learned to express the importance or weights of features in Vi to compute the final attention.

Next, the scaled dot-product attention tensor (A) is computed for every head. To achieve this, the SoftMax function is applied to the matrix multiplication between Ki and Qi, which is then divided by the square root of the head dimension. Each row in A represents a probability distribution of attention for a given query, indicating which keys (patches) are most similar to the query of that specific head.

The self-attention SA(z) is computed as the element-wise product between the matrices A and v. The SA(z) matrices are subsequently combined along the following dimension, resulting in a dimension of (n+1, d) tensor. Afterward, the tensor is processed by a single linear layer and undergoes element-wise multiplication with a learnable tensor. This linear layer plays a crucial role as it enables the learning of features as aggregates from all the attention heads, capturing comprehensive information from the different attention heads.

Lastly, MSA(z) represents the operation output applied to the input sequence z. $[SA_1(z); SA_2(z); \ldots SA_k(z)]$ represents the concatenation of the outputs from all the self-attention sub-layers and U_{msa} represents a learnable parameter matrix. Overall, this process of multi-head attention in the encoder enables the model to learn from diverse perspectives, compute attention weights, and aggregate features across all the heads.

4) Classification Head: In the vision transformer architecture, the final representation of the [cls] token serves as the basis for performing classification tasks. Before the trained procedure, a two-layer MLP is employed, resulting in multiple weight matrices: W_h with dimensions (d, d_{mlp}) and W_o with dimensions (d_{mlp}, d). These matrices capture the relationship between the representation and the classification labels. On the other hand, during fine-tuning, a single linear layer is utilized, leading to a tensor of dimension (d, n_cls), where n_cls represents the number of classes [26]. Regardless of the specific configuration, the network's output is a vector of size (1, n_cls), which consists of probabilities corresponding to each class, enabling classification based on the learned representation.

B. Random Projection

Random projection is a dimensionality reduction technique used in machine learning similar to Principal Component Analysis (PCA). It transforms high-dimensional data into a lower-dimensional space while preserving important structural information [29]. By using random projection, the computational complexity and memory requirements of a model can be significantly reduced without compromising the performance.

In the context of ViT model, random projection techniques are employed to reduce the dimensionality of the path embeddings. This makes the model more efficient and scalable, allowing for faster computation during training and inference. It enables the ViT model to handle large-scale image datasets more effectively. Moreover, random projection helps address the curse of dimensionality by mitigating the impact of high-dimensional feature spaces, leading to better generalization and improved performance in various computer vision tasks.

Given the set of original high-dimensional data, X, random projection can be expressed as $Y = X * R$, where Y represents the transformed data in the lower-dimensional space, and R is the random projection matrix. The random projection matrix, R, is typically a randomly generated matrix with dimensions determined by the desired reduction in dimensionality. Each element of R is drawn independently from a suitable probability distribution, such as a Gaussian distribution or a random uniform distribution. The elements of R can be real or complex numbers, depending on the nature of the data being transformed.

C. Randomized Vision Transformer (RViT)

The architecture of the proposed RViT model is depicted in Fig. 3.

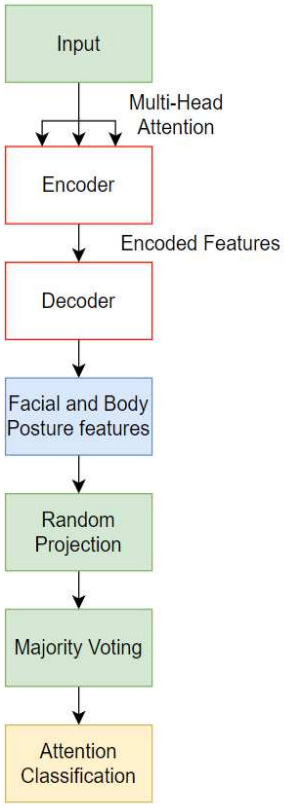

Fig. 3 Overview of RViT

The input data is processed through an encoder to capture important information and extract meaningful representations. Subsequently, a decoder refines these encoded representations to generate facial and body posture features. Afterward, randomized projection is applied to transform the encoded representation into a lower dimensional embedding to enhance computational efficiency. The projected features are then trained using the ViT model and combined through a majority voting scheme, incorporating diverse perspectives for decision-making. Finally, attention-based classification is employed, leveraging both the combined information and attention mechanisms to ensure accurate classification of the input data.

Majority voting is a commonly used technique in machine learning that combines the predictions of multiple models or classifiers to make decisions. It is especially useful in ensemble learning, where several models are trained independently on the same dataset. Each model provides its prediction for a given input, and the majority voting scheme combines these predictions to determine the final decision [30].

In binary classification tasks, the class label predicted by the majority of the model is selected. For example, if five classifiers predict Class A and three predict Class B, majority voting would assign the input to Class A. In multi-class classification, the class with the most votes is typically chosen. In this study, the implementation of majority voting was introduced to enhance the accuracy of the classification process. It addresses the issue identified in the binary classification tasks, where the individual images may be misclassified due to certain actions that can be confusing. For instance, an image of someone playing with their fingernails in the early stage might be incorrectly classified as late.

The majority voting mechanism functions by considering a sequence of images rather than just a single image. In our case, it classifies the concentration level based on every 200 predicted frames in a video. By analyzing the concentration levels of these frames, the system calculates the probabilities and makes a final conclusion. Once all frames in the video have been predicted, the system generates a final conclusion that indicates the student's concentration level.

Fig. 4 illustrates the majority voting process. In this example, there are a total of 10 images that the model has classified. Out of these images, 4 of them are classified as mid-stage, while the remaining classes each have two images. By identifying the classes with the highest number of predictions (the majority), the system concludes that the overall concentration level falls within the mid-stage category.

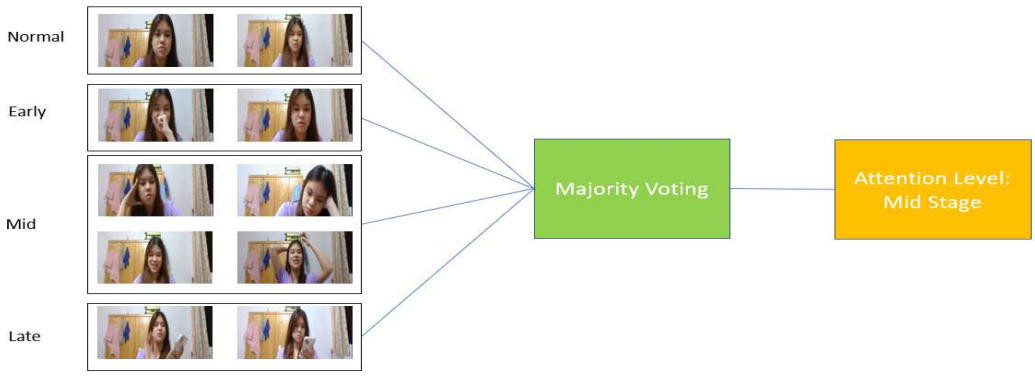

Fig. 4 Majority Voting Illustration

III. Result and Discussion

A. Data Collection

This study's self-collected database comprising video data from 20 participants is constructed, capturing various concentration levels in a classroom environment. The collected video datasets undergo pre-processing using video editing software to eliminate any unwanted sections in the video sequence, e.g., technical glitches during the preparation of recording software or when the video recording session becomes idle due to issues with the internet connection. Furthermore, various pre-processing techniques, such as cropping, exposure adjustment, and blur removal, are

employed to enhance the video quality. This is necessary as the videos of each participant were captured in diverse environments containing different backgrounds and lighting conditions.

A collection of 35,151 image frames from various video sequences was captured and saved in /jpeg format. Each image was assigned a name according to its class, such as normal586, early174, mid937, and late 496. The images belonging to different classes were organized into separate folders, which were later extracted and divided into training and testing folders. Specifically, there are 3,721 images for the normal stage, 7,624 for the early stage, 11,673 for the middle stage, and 12,133 for the late stage. However, due to limited computer resources, only 7,000 images were utilized in this experiment. The training set consisted of 4,000 images, while the testing set contained 3,000 images. Fig. 5 and Table II present the experiment dataset information in graphical and tabular format and a selection of sample images representing each class.

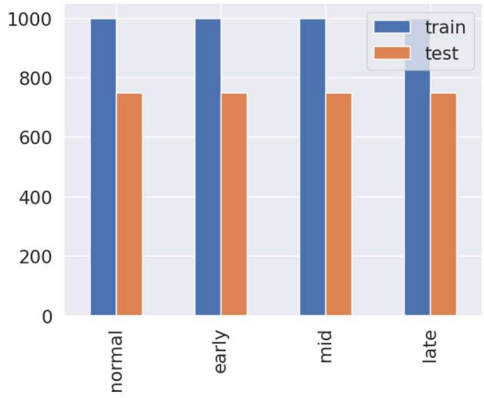

Fig. 5 Number of training and testing images in the dataset

Fig. 6 displays some sample images representing each concentration stage. In the normal stage, the participants do not exhibit any unnecessary movements. Their heads are aligned with their lines of sight, and they maintain a steady and direct eye contact with the screen. During the early stage, the participants avoid direct contact with the screen and display less frequent glances to the side. However, they are still actively trying to concentrate, as evidenced by the detection of small movements. Furthermore, various small actions, such as tapping fingers and playing with fingernails, are observed. These actions may suggest a certain level of restlessness or nervousness while the participants maintain their focus on the task at hand.

Moving to the mid stage, the participants' eyes frequently shift to the side, such as continuously checking the time on their smartphones, but their gaze consistently returns to the screen. Some participants' heads start to glance around and are often supported by their hands. They also exhibit frequent shifts in body posture, as they struggle to remain still.

Finally, in the late stage, the participants completely lose focus. They avoid eye contact with the screen entirely, and their heads frequently turn around as if searching for something more interesting. Some participants even engage in activities such as playing with or talking on their phones in front of the screen.

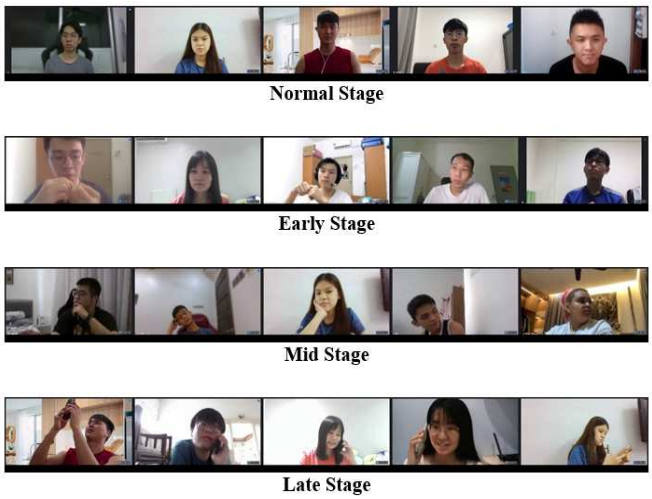

Fig. 6 Sample Datasets

TABLE II
SUMMARY OF DATASET

Stage	Normal	Early	Mid	Late	Total
Samples	3,721	7,624	11,673	12,133	35,151
Training Samples	1,000	1,000	1,000	1,000	4,000
Testing Samples	750	750	750	750	3000

B. *Evaluation Metrics*

Several evaluation metrics are used to assess the performance of the proposed method. They provide standardized ways to evaluate and compare the different configurations of the approach. The evaluation metrics used in this study include:

1) Confusion Matrix: to determine the true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

2) Precision: to measure the accuracy of positive predictions, considering both true positive and false positive.

3) Recall – to measure the accuracy of identifying positives, including true positives and false negatives, relative to the actual number of positives.

4) F1-Score: a combination of precision and recall to measure the accuracy and ability of a model in correctly identifying positive instances.

5) Training Loss: to measure the model's error on the training data, indicating how well it matches the training set.

6) Validation Loss: similar to training loss, to evaluate a model's performance on a separate validation set by calculating the sum of errors for each sample.

C. *Performance of the Proposed RViT Method*

Various hyperparameters are experimented with to achieve the optimal detection rate for the dataset. The following summary outlines the performance of each model using different hyperparameters. In general, assigning a higher number of training epochs to the algorithm leads to improved model performance, as it allows the model's weights to better adapt to the dataset. In this section, different batch sizes were tested for the model. Consequently, the model's name

indicates the sequence of the batch size used, such as RViT_16.

Table III illustrates the training loss and accuracy of RViT models with different batch sizes. After 50 training epochs on the dataset, these models show slightly lower performance than those without random projection. This discrepancy in performance could be attributed to the potential loss of information during the implementation of random projection. However, the accuracy of RViT_64 has improved to 90.80%, making it the highest accuracy among the models. Moreover, it is noticeable that the difference between the accuracy and validation accuracy of each model is small, suggesting that the models exhibit decent generalization capabilities.

The learning history of the normal ViT model with batch size of 64 (ViT_64) and RViT_64, is presented in Fig. 7. This compares each model's generalization capabilities and determines whether overfitting occurred during the model training. Based on the results, we can see that although ViT_64 (first figure) has an accuracy of 90% above, it is slightly overfitted as it has a huge difference compared to its validation accuracy.

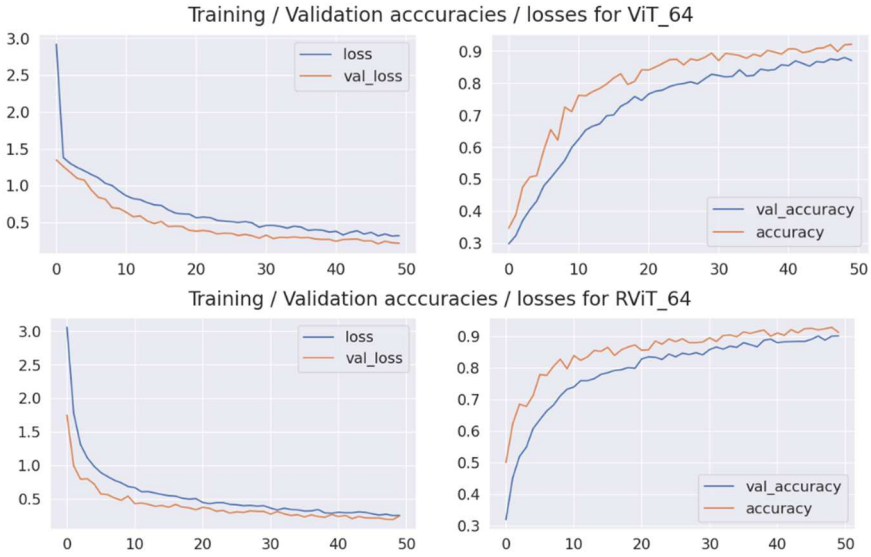

Fig. 7 Learning History of ViT_64 and RViT_64

On the other hand, the smooth loss graph in RViT_64 (second figure) suggests that the model is converging steadily and steadily improving throughout the training process. Additionally, the accuracy and validation accuracy curves nearly coincide, indicating that the model achieves a balanced performance without overfitting the training data. This demonstrates the model's ability to generalize well and make accurate predictions on unseen data.

TABLE III
PERFORMANCE RESULT OF RVIT

Model	Loss	Accuracy	Validation Loss	Validation Accuracy
RViT_16	0.3301	0.8758	0.2694	0.9040
RViT_32	0.2821	0.8945	0.2903	0.8947
RViT_64	0.2540	0.9080	0.2725	0.9057

D. Result of RViT with Different Randomized Hyperplanes

In this section, we evaluate the effects of using different randomized hyperplanes, specifically the number applied when performing random projection on the layers extracted prior to the fully connected layer. Altering the different number of randomized hyperplanes results in higher or lower dimensional outputs. This can enable the model to learn more or less complex representations and capture varying numbers of fine-grained features in the data. The model that demonstrated the best performance in Section 4.3 is selected in the test.

Each model is named in this experiment based on the assigned random projection value. For instance, RViT_R32 indicates that the number of random hyperplanes is set to 32. Table IV displays the performance outcomes of RViT with different hyperplane numbers. The results indicate that RViT_R64, RViT_R128, and RViT_R256 exhibit nearly identical accuracy scores of 90.80%, 90.20%, and 90.50%, respectively. RViT_R32, on the other hand, demonstrates the highest loss value during training, while RViT_R128 achieves the highest validation accuracy. Based on this information, it can be concluded that having 64 randomized hyperplanes is the most suitable for the concentration level detection problem.

TABLE IV
PERFORMANCE WITH DIFFERENT RANDOM PROJECTION VALUES

Model	Loss	Accuracy	Validation Loss	Validation Accuracy
RViT_R32	0.2849	0.8985	0.2714	0.9000
RViT_R64	0.2540	0.9080	0.2725	0.9057
RViT_R128	0.2649	0.9020	0.1996	0.9353
RViT_R256	0.2713	0.9050	0.2078	0.9230

E. Error Analysis

Fig. 8 illustrates some examples of misclassified images from model RViT_64. These misclassifications primarily arise from participant behaviors that can mislead the model's classification. For instance, participants often exhibit patterns where they initially focus on the screen during the normal and early stages, but avoid looking at the screen during the mid and late stages. Consequently, if the model detects that a

participant is not gazing at the screen, it tends to classify them as being in either the mid or late stage.

Fig. 8 Misclassified Images

However, there are cases where participants in the early stage do not glance at the screen. In such situations, participants who exhibit fidgeting behavior while not looking at the screen can be misclassified by the model as being in the mid or late stage. This highlights the challenges faced in accurately classifying participants based on their gaze behavior, particularly when certain actions or circumstances deviate from the expected patterns.

F. Time and Computing Resource Evaluation

This section examines the training time and the computational demand difference between RViT and ViT models without applying randomized projection. It is hypothesized that models with reduced dimensions, achieved through randomized projection, require less training time and computational demands.

Table V presents the overall training time for each model, considering both models with and without the implementation of random projection. The results confirm our hypothesis, as the models incorporating random projection exhibit reduced training durations while still can maintain a noteworthy performance level.

TABLE V
SUMMARY OF MODELS' TRAINING DURATION

Batch Size	Learning Rate	Total Training Duration	
		ViT	RViT
16	0.001	747	649
32		572	483
64		493	485

Fig. 9 illustrates the disparities in computing resource utilization between training the ViT_64 and RViT_64 models. Notably, there is a substantial contrast in the usage of system RAM and GPU RAM. While training the ViT_64 model pushed the system RAM to its limits, approximately 2GB of available system RAM remained after training the RViT_64 model. Furthermore, the GPU RAM usage was higher for the ViT_64 model than the RViT_64 model. Additionally, due to the larger data dimensions, disk usage also increased. These findings support our hypothesis that models with reduced dimensions necessitate fewer computing resources for training. However, it is important to consider that training the models on different hardware configurations may result in varying resource requirements, and employing higher-spec hardware may lead to reduced computing resource usage.

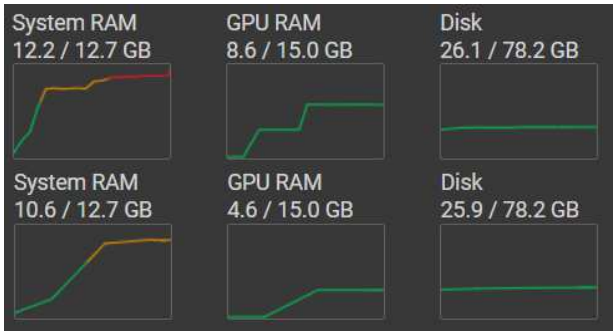

Fig. 9 Resources usage of ViT_64 and RViT_64

G. Comparisons with State-of-the-Art Techniques

In this section, we compare the proposed approach with state-of-the-art techniques. Since RViT is a deep learning model, we will compare our proposed work with existing deep learning methods. The RViT_64 model is chosen in this study.

Table VI summarizes the performance outcomes attained by different authors' approaches, as mentioned in Section 2, in contrast to RViT_64. When compared to Canedo et al.'s work, these methods achieve an accuracy of 85% in determining students' concentration levels. However, the analysis of the experimental results is restricted, and there is limited elaboration on the training and testing procedures in those studies.

TABLE VI
COMPARISON WITH OTHER METHODS

Methods	Classifier	Dataset	Accuracy
Canedo *et al.* (2018) [17]	CNN	Self-collected dataset	85%
Li *et al.* (2020) [18]	CNN	Self-collected dataset	85.3%
Revadekar *et al.* (2020) [3]	CNN	Self-collected dataset	99.82%
Shamika *et al.* (2021) [20]	CNN	DAISEE dataset	90.6%
RViT_64 (proposed)	ViT	Self-collected dataset	90.80%

On the other hand, Li et al. introduce an approach that considers students' facial attitudes to assess their concentration levels. However, students tend to seek out interesting stimuli when they feel bored, leading to frequent changes in body postures. As a result, our proposed work differs by incorporating not only facial attitude and expression but also the students' body posture as indicators of concentration levels.

Regarding Revadekar et al.'s study, they determine concentration levels using students' facial expressions and body postures, which aligns with our own work. The authors assert an accuracy of 99.82%; however, they have not evaluated the model's generalization ability. Consequently, there are concerns that their proposed work may be overfitted due to the remarkably high accuracy achieved. Additionally, there is a lack of adequate information regarding the analysis of the results of their study.

Finally, our model exhibits comparable accuracy and generalization capabilities to the method proposed by Shamika et al. However, the authors utilize a large dataset, resulting in significantly longer training times of approximately 275 seconds per epoch on average. In contrast, our approach achieves similar results within a much shorter time frame, with an average epoch duration of only 9 seconds.

Our proposed work takes into account the limitations observed in the studies above by incorporating additional factors to determine concentration levels. We address the issue of insufficient result analysis by conducting a comprehensive analysis of our experimental outcomes. Furthermore, we consider a broader range of features, including facial attitude, expression, and body posture, to provide a more comprehensive understanding of students' concentration levels. Another significant improvement in our approach is the reduction in training time compared to Shamika et al.'s method. We have achieved comparable results within a significantly shorter timeframe by employing optimized algorithms and leveraging efficient computational techniques. This reduction in training time is crucial in practical applications, as it allows for quicker model development and deployment, enhancing our proposed solution's overall efficiency and usability.

IV. Conclusion

Detecting student concentration is a critical aspect that profoundly influences the learning process. To address this, different analyses and comparisons with the current existing work have been conducted, and various techniques combining machine learning and computer vision have been explored to identify the most effective and reliable algorithm for real-world applications. Among the different models tested with various hyperparameters, the RViT model with a batch size of 64 has emerged as the most promising. This algorithm demonstrates exceptional capabilities in generalization and requires less time and resources for training compared to other alternatives.

In our future studies, we plan to expand the scope of our experiments by including a larger dataset. Due to hardware limitations in the current experiment, we could not incorporate all the collected data. Additionally, we aim to explore alternative techniques, such as applying SVM and KNN algorithms, after reducing the ViT model using random projection. We will also consider integrating other methodologies like OpenPose and OpenFace for more comprehensive analysis. By incorporating these advancements, we anticipate that our algorithm will offer a more comprehensive and reliable estimation of the concentration state.

Acknowledgment

This research is supported by the Fundamental Research Grant Scheme (FRGS/1/2020/ICT02/MMU/02/5) and MMU IR Fund (MMUI/220026).

References

[1] R. Shafique, W. Aljedaani, F. Rustam, E. Lee, A. Mehmood, and G. S. Choi, "Role of Artificial Intelligence in Online Education: A Systematic Mapping Study," *IEEE Access*, vol. 11, pp. 52570–52584, 2023, doi: 10.1109/ACCESS.2023.3278590.

[2] Y. Shi, F. Sun, H. Zuo, and F. Peng, "Analysis of Learning Behavior Characteristics and Prediction of Learning Effect for Improving College Students' Information Literacy Based on Machine Learning," *IEEE Access*, vol. 11, pp. 50447–50461, 2023, doi: 10.1109/ACCESS.2023.3278370.

[3] A. Revadekar, S. Oak, A. Gadekar, and P. Bide, "Gauging attention of students in an e-learning environment," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, Dec. 2020, pp. 1–6. doi: 10.1109/CICT51604.2020.9312048.

[4] D. M. Cretu and Y.-S. Ho, "The Impact of COVID-19 on Educational Research: A Bibliometric Analysis," *Sustainability*, vol. 15, no. 6, Art. no. 6, Jan. 2023, doi: 10.3390/su15065219.

[5] A. Kumar *et al.*, "Impact of the COVID-19 pandemic on teaching and learning in health professional education: a mixed methods study protocol," *BMC Medical Education*, vol. 21, no. 1, p. 439, Aug. 2021, doi: 10.1186/s12909-021-02871-w.

[6] B. Meriem, H. Benlahmar, M. A. Naji, E. Sanaa, and K. Wijdane, "Determine the Level of Concentration of Students in Real Time from their Facial Expressions," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 1, Art. no. 1, 55/31 2022, doi: 10.14569/IJACSA.2022.0130119.

[7] G. J. DuPaul, P. L. Morgan, G. Farkas, M. M. Hillemeier, and S. Maczuga, "Academic and Social Functioning Associated with Attention-Deficit/Hyperactivity Disorder: Latent Class Analyses of Trajectories from Kindergarten to Fifth Grade," *J Abnorm Child Psychol*, vol. 44, no. 7, pp. 1425–1438, Oct. 2016, doi: 10.1007/s10802-016-0126-z.

[8] S. T. Lim, J. Y. Yuan, K. W. Khaw, and X. Chew, "Predicting Travel Insurance Purchases in an Insurance Firm through Machine Learning Methods after COVID-19," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, Art. no. 2, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.4.

[9] S. V. Mahadevkar *et al.*, "A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions," *IEEE Access*, vol. 10, pp. 107293–107329, 2022, doi: 10.1109/ACCESS.2022.3209825.

[10] M.-C. Su, C.-T. Cheng, M.-C. Chang, and Y.-Z. Hsieh, "A Video Analytic In-Class Student Concentration Monitoring System," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 4, pp. 294–304, Nov. 2021, doi: 10.1109/TCE.2021.3126877.

[11] M. M. A. Parambil, L. Ali, F. Alnajjar, and M. Gochoo, "Smart Classroom: A Deep Learning Approach towards Attention Assessment through Class Behavior Detection," in *2022 Advances in Science and Engineering Technology International Conferences (ASET)*, Feb. 2022, pp. 1–6. doi: 10.1109/ASET53988.2022.9735018.

[12] J. Zaletelj and A. Košir, "Predicting students' attention in the classroom from Kinect facial and body features," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 80, Dec. 2017, doi: 10.1186/s13640-017-0228-8.

[13] C. Thomas and D. B. Jayagopi, "Predicting student engagement in classrooms using facial behavioral cues," in *Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education*, in MIE 2017. New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 33–40. doi: 10.1145/3139513.3139514.

[14] X. Zhang, C.-W. Wu, P. Fournier-Viger, L.-D. Van, and Y.-C. Tseng, "Analyzing students' attention in class using wearable devices," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2017, pp. 1–9. doi: 10.1109/WoWMoM.2017.7974306.

[15] Q. Deng and Z. Wu, "Students' Attention Assessment in eLearning based on Machine Learning," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 199, no. 3, p. 032042, Dec. 2018, doi: 10.1088/1755-1315/199/3/032042.

[16] N. Veliyath, P. De, A. A. Allen, C. B. Hodges, and A. Mitra, "Modeling Students' Attention in the Classroom using Eyetrackers," in *Proceedings of the 2019 ACM Southeast Conference*, in ACM SE '19. New York, NY, USA: Association for Computing Machinery, Apr. 2019, pp. 2–9. doi: 10.1145/3299815.3314424.

[17] D. Canedo, A. Trifan, and A. J. R. Neves, "Monitoring Students' Attention in a Classroom Through Computer Vision," in *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, vol. 887, J. Bajo, J. M. Corchado, E. M. Navarro Martínez, E. Osaba Icedo, P. Mathieu, P. Hoffa-Dąbrowska, E. Del Val, S. Giroux, A. J. M. Castro, N. Sánchez-Pi, V. Julián, R. A. Silveira, A. Fernández, R. Unland, and R. Fuentes-Fernández, Eds., in Communications in Computer and Information

Science, vol. 887. , Cham: Springer International Publishing, 2018, pp. 371–378. doi: 10.1007/978-3-319-94779-2_32.

[18] S. Li, Y. Dai, K. Hirota, and Z. Zuo, "A Students' Concentration Evaluation Algorithm Based on Facial Attitude Recognition via Classroom Surveillance Video," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 24, no. 7, pp. 891–899, 2020, doi: 10.20965/jaciii.2020.p0891.

[19] J. N. Mindoro, N. U. Pilueta, Y. D. Austria, L. Lolong Lacatan, and R. M. Dellosa, "Capturing Students' Attention Through Visible Behavior: A Prediction Utilizing YOLOv3 Approach," in *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, Aug. 2020, pp. 328–333. doi: 10.1109/ICSGRC49013.2020.9232659.

[20] U. B. P. Shamika, W. A. C. Weerakoon, P. K. P. G. Panduwawala, and K. A. P. Dilanka, "Student concentration level monitoring system based on deep convolutional neural network," in *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, Sep. 2021, pp. 119–123. doi: 10.1109/SCSE53661.2021.9568328.

[21] K. Han *et al.*, "A Survey on Vision Transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.

[22] X. Yu, J. Wang, Y. Zhao, and Y. Gao, "Mix-ViT: Mixing attentive vision transformer for ultra-fine-grained visual categorization," *Pattern Recognition*, vol. 135, p. 109131, Mar. 2023, doi: 10.1016/j.patcog.2022.109131.

[23] W. Sun *et al.*, "Vicinity Vision Transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12635–12649, Oct. 2023, doi: 10.1109/TPAMI.2023.3285569.

[24] Y. Lou, R. Wu, J. Li, L. Wang, X. Li, and G. Chen, "A Learning Convolutional Neural Network Approach for Network Robustness Prediction," IEEE Transactions on Cybernetics, vol. 53, no. 7, pp. 4531–4544, Jul. 2023, doi: 10.1109/tcyb.2022.3207878.

[25] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[26] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv, Jun. 03, 2021. doi: 10.48550/arXiv.2010.11929.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, May 24, 2019. doi: 10.48550/arXiv.1810.04805.

[28] A. Vaswani *et al.*, "Attention Is All You Need." arXiv, Aug. 01, 2023. doi: 10.48550/arXiv.1706.03762.

[29] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Johnson-Lindenstrauss Lemma, Linear and Nonlinear Random Projections, Random Fourier Features, and Random Kitchen Sinks: Tutorial and Survey." arXiv, Aug. 09, 2021. doi: 10.48550/arXiv.2108.04172.

[30] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, Sep. 1997, doi: 10.1109/3468.618255.