Quantifying Students' Involvement during Virtual Classrooms: A Meeting Wrapper for the Teachers

Snigdha Das, Sandip Chakraborty, Bivas Mitra
Department. of Computer Science and Engineering, Indian Institute of Technology Kharagpur
India
snigdhadas@iitkgp.ac.in,{sandipc,bivas}@cse.iitkgp.ac.in

ABSTRACT

Educational systems have witnessed a rapid transformation from the traditional physical classroom to the online teaching mode due to the COVID-19 pandemic. This sudden shift throws two significant questions to the educational policymakers and system designers. (1) How can an instructor improve the teaching performance over an online teaching platform? (2) How does the instructor understand the students' learning pace? For answering these questions, we propose a platform in this paper that analyzes the real-time presentation video and the facial video feeds from the instructor as well as the students to explore the visual engagement of the student towards the lecture contents. However, this is challenging as the students may get involved in various multitasking instances, such as taking notes or browsing relevant reading materials. The crux of this paper is to understand a few real-time opportunistic moments when the students should visually focus on the presentation content if they are engaged and been able to follow the lecture. We investigate these instances and analyze the visual engagement of the students from their eye gaze and gaze gestures in real-time during those instances to assess their engagement during an online class. Our system achieves 71% (standard deviation 10%) accuracy for all over the scenarios in students' involvement detection during the virtual classroom.

KEYWORDS

online lecture, attention, engagement, video session

ACM Reference Format:
Snigdha Das, Sandip Chakraborty, Bivas Mitra. 2021. Quantifying Students' Involvement during Virtual Classrooms: A Meeting Wrapper for the Teachers. In *India HCI 2021 (India HCI 2021), November 19–21, 2021, Virtual Event, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3506469.3506492

1 INTRODUCTION

Apart from teaching, understanding the students' progress and learning ability is one of the primal roles of an instructor in a class. Proper knowledge about a student's learning status is needed not only to assist the student in coping with the course but also to improve the instructors' presentation skills. In an in-person physical classroom, the teacher or the instructor can continuously monitor the body and eye gestures of the students; such physical eye contact between the instructor and the students can provide significant cognitive sources of information for assessing the students' learning pace in real-time. However, the COVID-19 pandemic has forced instructors to conduct the classes virtually through various online meeting platforms. Unfortunately for different Afro-Asian countries that belong to the low and middle-economy classes, virtual classrooms are infancy for a majority of the instructors. Various recent reports[1] have indicated that a majority of the instructors, particularly the school teachers and the college lecturers from suburban and rural areas, are not accustomed to the online mode of teaching. Various limitations of the online meeting platforms exaggerate this problem further. For instance, the meeting platforms can only show the video feeds from hardly 4-8 students, whereas a typical class size is more than 50, sometimes it crosses 100. Consequently, a teacher can never feel the cognitive connectivity with the students, which is very much essential in any teaching-learning process [2, 4, 11].

Interestingly, a few works in the literature [5, 13] discourse the necessity of the instructor's enhancement in the teaching process, which addressed the issues in the teaching system and proposed various strategies for instructor's enhancement over a physical mode of education. On the other side, for understanding the students' learning pace, various recent studies have advocated using specialized wearable devices [9, 12] such as wearable eye trackers, smart glasses, and smartphone sensors, etc. for capturing the attention information through eye dynamics and physiological signature. However, such specialized devices are never a feasible option for the low and middle economy Afro-Asian countries. A second notion of works consider different student engagement detection schemes [6, 8, 10, 14] by quantifying the visual engagement [3, 6] of the students over the meeting application. These works primarily measure a student's attentiveness by quantifying the amount of time they gazed on the device's screen that shows the meeting platform. However, visual engagement gives a very crude idea about the cognitive attentiveness of a student. During the classes, the students can perform various other related activities, like taking notes, browsing relevant materials over the Internet, etc., which certainly boosts their cognitive attention towards the class but reduces their visual engagement.

The crux of this paper is as follows. Although visual engagement gives a crude idea about the cognitive attentiveness of a student, it is

India HCI 2021, November 19–21, 2021, Virtual Event, India

ACM ISBN 978-1-4503-9607-3/21/11…$15.00
https://doi.org/10.1145/3506469.3506492

[1] https://thelogicalindian.com/education/legrand-30615?infinitescroll=1 (Accessed: January 6, 2022)

inevitable during certain portions of the lectures, particularly when the instructor explicitly seeks the visual attention from the student. For example, when the instructor explains a diagram or displays certain animations related to the explanation, it is more likely that the student would focus on the screen. Our proposed method investigates this idea and finds opportunities to infer whether the students are sufficiently attentive to the lecture during the class hour. Accordingly, we analyze the local video feeds from the students and the instructor to map the visual engagement to the cognitive attentiveness correctly. In our method, the video feed of the instructor, which particularly contains the presentation video along with the instructors' facial feeds, is used as the benchmark to extract the opportunities when a visual engagement from the students is necessary. The method then analyzes the video feed of the students to find out whether the students show a visual cue towards the screen and accordingly generates an engagement score for the students. We develop a prototype of this method and evaluate it in two different types of real-time online lecture sessions – online classroom teaching and group discussions with a presentation (total of 4 sessions, 7 – 12 participants). Our system achieves 71% (standard deviation 10%) accuracy for all over the scenarios.

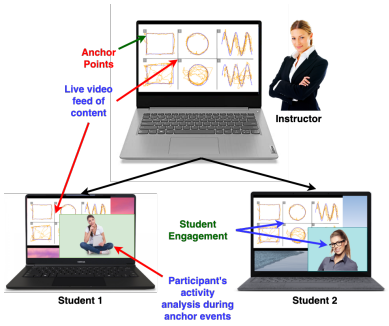

Figure 1: Instructor and students' system overview for detecting student engagement

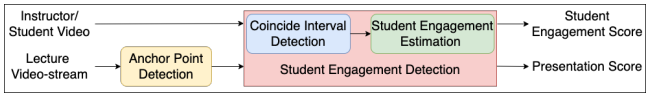

Figure 2: Student engagement detection framework

2 SYSTEM OVERVIEW

Figure 1 shows an overview of the student engagement detection system in an online teaching platform where each student and the instructor use webcam-based system for virtual classrooms. The proposed system operates using three major components – (1) the instructor's presentation video, (2) the instructor's facial video feed, and (3) the students' facial video feed. The former two components work as the benchmark to determine opportunistic events when visual engagement from the students is necessary. We term these opportunistic instances as the "*Instructor's Understandability*" because it determines the benchmark for the instructors to get a perspective about the students' attentiveness. The related events on the presentation, like an animation, are termed as the "*Anchor Points*". In contrast, the first and last components jointly find out the visual engagement of the student in the front screen during those opportunistic events. Finally, the two later components decide the student's engagement in an online class. While the instructor's understandability and the student's engagement in the front screen are performed locally at devices on which the meeting application runs, the concluding student engagement in the online class is processed at the instructor's device using the students' video meta-data shared over the platform. Figure 2 outlines the student engagement detection framework which is primarily composed of two modules – (a) *Anchor Point Extraction*, and (b) *Student Engagement Detection*. The former module captures the transitions in the instructor's presentation video. In contrast, the latter concentrates on the instructor's understandability and the student engagement detection in the front screen and the online class. In summary, all the videos of the students are processed at their end and the extracted information is shared with the instructor for quantifying the students' involvement.

(a) Anchor Point Extraction: This module runs both the instructor and the student's device and excerpts the anchor points from the presentation video by analyzing the transition in the consecutive frames of that video. Each presentation consists of significant slides and a few insignificant ones that do not contain any demonstrating points. We utilize the fact at the beginning of the processing to eliminate such insignificant slides so that the computation power for running the process is minimized. Furthermore, the significant slides may contain insignificant content, such as very text-heavy content or a small change in the slide. We further remove these insignificant contents for extracting the anchor events. (Details in **Section 3**).

(b) Student Engagement Detection: Partially, this module executes on both the instructor and the students' devices, and the rest runs on the instructor side to generate the student engagement scores during the anchor events. The instructor and the engaged students pursue the presentation video at least during the most significant anchor events. Hence, the eye gaze changes with transition in the video frames. We utilize this information to identify the good presenter (who follows the presentation) and the students engaged at the front screen. Furthermore, by utilizing the fact, we find the similarity between the instructor and the student during the anchor events. But the similarity measure is not trivial as both the video feed of the instructor and the students are asynchronous. Moreover, the instructor view is unavailable to the students as it requires multiple object tracking from the student side. For dealing with the asynchronous video feed, a delay component is added. Additionally, as there is no direct view available, a discrete point comparison mechanism is used to compare the eye movement of the instructor and the student for generating the student engagement score. (Details in **Section 4**)

3 ANCHOR POINT EXTRACTION

This section describes the steps for extracting the anchor points from the presentation video. The presentation video is processed locally at the instructor and the students' system. In an online teaching system, the instructor and the students use the same platform for sharing and viewing the presentation document and the

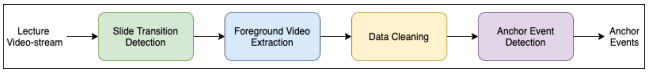

Figure 3: Anchor Point Detection Framework

Figure 4: Initial Slide

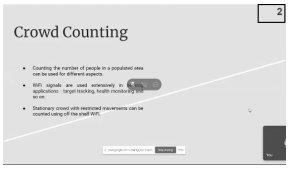

Figure 5: Consecutive Slide

video feed. Therefore, we consider that all presentation sessions are streamed at a rate of f fps. Depending on the study outcomes, the presentation slide is selected as the teaching mode. The overall framework of anchor point detection is shown in Figure 3.

3.1 Slide Transition Detection

In the presentation slide-based teaching, several insignificant slides such as starting, ending, and title slides are present. In this module, we partition the videos to eliminate such insignificant slides. For segmenting the significant video slices, we rely entirely on the slide number of the presentation. The slide number is present in all the slides except for the insignificant one. Therefore, we first locate the slide number position in the slide. The slide number typically is spotted either in the upper right corner, lower right corner, or middle bottom of the slide. During the slide transition, the pixel values of either of the three portions reasonably change, and the rest two remain the same.

To detect the slide transition, we apply a 30 pixels grid on the three pre-defined positions of each video frame to crop the portion containing the slide number. The starting slide with no slide number is treated as the initial template for matching the subsequent slides. After retrieving the cropped frame portions, we convert that into a greyscale image and compare each cropped image with the respective cropped image of the subsequent frame using mean squared error. During the slide transition, only comparing the cropped images with slide numbers produces a high difference value. In contrast, the rest of the portion comparison in transition or non-transition comparison generates almost zero difference value. Hence, we apply a simple threshold value, δ, to slice the slides. Figures 4 and 5 show the greyscale images of initial and the next slide, respectively. The comparison of the top right corner of the two images contributes to a larger pixel difference, thus concluding the detection of the slide transition event.

3.2 Foreground Video Extraction

Once the slide is detected, our next task is to identify the object movement within the slide. In the academic presentation, the entire presentation consists of a single template. Hence, the slide template acts as the background of the video for the whole of the presentation. The varying presentation content appears as the foreground

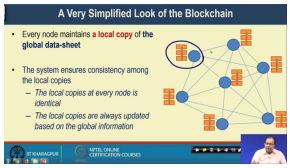

Figure 6: Lecture video feed at the student end

Figure 7: Foreground content of the presentation video frame

substance of the video. Therefore, the slides with invariant content display a regular pixel pattern. Identification of such patterns helps split out the presentation content containing the irregular pattern. This process is analogous to the detection of the intruding object from the image. Similar to the background scene of the intruder object, the regular pattern of the presentation template can be expressed by the statistical model, and without following the model is indicated as varying presentation content. Thus, we apply the background subtraction method for separating the background template and foreground content.

We use the Gaussian mixture model-based background subtraction method [17] for identifying the mixture characteristics of the background template and extracting the foreground movable presentation content. In particular, each pixel of the frame is described by a mixture of Gaussian as the presentation video may be impacted by sudden lighting change effect, addition and removal of content, and slow-moving content. By learning the variance of each of the Gaussian of the mixture, the model determines the likeliness of the Gaussian as a background pixel. The continuous learning process includes the recent background information for adapting to the current changes in the frame. Specifically, we look prior 500 frames of the current frame to estimate the next frame's pixel trait. Finally, the pixel values that do not map with any of the background distributions are considered foreground. The extracted foreground information containing the varying presentation content is kept as a binary formatted image.

3.3 Data Cleaning

The foreground video extraction module's outcome contains the presentation content and a few scattered points that occur due to the imprecise learning of the Gaussian parameters. In such a scenario, we need an image smoothing model to eliminate the scattered points on the foreground video frame. The general strategy of image smoothing is to act as a low-pass filtering kernel. However, we have binary formatted foreground video, containing either of 0 and 255 values. Hence, statistical computation-based filtering such as mean and Gaussian filtering, which may generate a different pixel value, is not suitable. Therefore, we select median filtering on the foreground video to remove such salt pepper random points. Figures 6 and 7 show the presentation video frame and the extracted foreground content, respectively.

3.4 Anchor Event Detection

Once we receive the filtered foreground video, our final job is to identify the anchor point from the varying presentation content. By

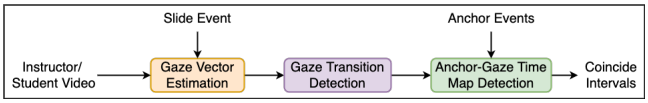

Figure 8: Coincide Interval Detection Framework

anchor point, we especially define animation, image, highlighted, and short text. However, the filtered foreground video contains not only the anchor points but also the bulky text information. Furthermore, for registering the change in the human mind, the movable content must try at the scene for a certain period. Hence, we need another layer of filtering in addition to the data cleaning module. For encountering the anchor events, a Spatio-temporal threshold method is applied. The spatial threshold is required to check bulky or insignificant changes in the presentation video, whereas the temporal threshold is needed to eliminate irrelevant non-resistant content. Finally, we mark an event as an anchor event when the foreground frame pixel count is within the spacial thresholds δ_{s_1} and δ_{s_2} and that spacial constraint persists at least for δ_t number of frames where δ_t is the temporal threshold. Any violation of the Spatio-temporal threshold marks the foreground selection as the non-anchor event.

4 STUDENT ENGAGEMENT DETECTION

The final module, student engagement detection, comprises two sub-modules – (1) coincide interval detection and (2) student engagement estimation. The first submodule deals with mapping the student video with the presentation one during the priorly detected anchor event, and we call the mapping interval the coinciding interval. The last submodule estimates the student engagement using the student and instructor video feed during the coincide interval.

4.1 Coincide Interval Detection

Once we trace the anchor events in the presentation video, our next job is to analyze the students and the instructor's video to identify whether the person is looking at the screen. The outcome of the submodule is two folds. Firstly, the instructor looking at the screen reveals that they are following the presentation as we assume that the instructor is performing a single task during the online presentation session. Hence, they are a good presenter and acceptable for further student engagement processing. However, the students may perform multiple activities during the presentation session. Therefore, our second finding is that the student looking at the screen may follow the presentation and be eligible for the next phase of the engagement detection. The overall coincide interval detection framework is shown in Figure 8.

4.1.1 Gaze Vector Estimation. For detecting a person looking at the screen, our primal task is to monitor the person's video feed captured through the front camera associated with the presentation running device. Specifically, if a person looks at the screen, their eye is visible to the video feed. Hence, we estimate the eye gaze to determine the person's presence in the session. Instead of processing the complete video, we choose the person's video feed for the duration, which is the same as the slide slice videos from the outcome of the slide transition module (Section 3.1). Due to the selection of only the significant slides' execution time, our computation cost for gaze vector estimation is reduced. Once we receive the slide event from the presentation videos, we segment the person's video feed and process the individual video feed accordingly.

We use an appearance-based model [15] for detecting the gaze vector from the person's video. As part of the pre-processing, the model first detects the face and eye landmarks for each video frame. Then, it normalizes each frame to remove any kind of capture issues. Finally, it feeds the head and the eye information to the multimodal CNN model, which is trained on the MPIIGaze dataset [16] for estimating the gaze vector. For each frame, we receive a 3D gaze direction vector and consider only the gaze vector along X and Y axes as there exists no such movement along the Z-axis while attending the presentation. Hence, we pass $< frame, gaze_x, gaze_y >$ to the consecutive module for each frame.

4.1.2 Gaze Transition Detection. In this component, we capture the gaze direction vector transition during each of the slide slice videos. Our primal intuition is that the gaze direction abruptly changes in the presentation session depending on the presentation content. Moreover, the standard statistical methods are suitable for capturing the actual transition [1]. Hence, we rely on the change point detection methods for detecting the gaze transition. We use Pruned Exact Linear Time (Pelt) [7] change point detection method because of three-way benefits. Firstly, it works without knowing the number of change points present in the system. Secondly, it runs in linear time and ends up with an optimal result. Thirdly, the method is unsupervised. It runs on a mechanism that minimizes the penalized cost when considering a change point in the system. We use a non-parametric kernel-based cost function, *rbf*. As the gaze vector is a multi-variant, we apply a 2-norm on the gaze vector to make it uni-variant. Finally, after computing the change points, we receive a set of frame numbers where the gaze transition occurs.

4.1.3 Anchor-Gaze Time Map Detection. The final component of coincide interval detection module comprises of the mapping between the anchor events detected from the *Anchor point extraction* module (Section 3) and *Gaze transition detection* (Section 4.1.2). Our insight is that both the instructor and the student following the presentation session must look at the screen during the anchor event. However, in a real-time scenario, a delay may occur due to technological issues. Therefore, we have set up a delay bound of d frames where the instructor and the student look at the screen within d frames after completing the anchor event. Looking at the screen refers to the change in direction in the gaze direction vector. Within a slide, if the instructor's gaze transitions overlap with anchor events, we mark that instructor as *good presenter* and continue to the next engagement estimation module with the overlapping frame details. Otherwise, we inform the instructor that their eye is not following the presentation while presenting. On the other hand, if the overlap is present in the student's gaze transition and anchor event within a slide event, we track the frame number and pass that to the next module. In the absence of any overlap, we mark the student as *non-engaged* and exclude them from the current slide for further processing.

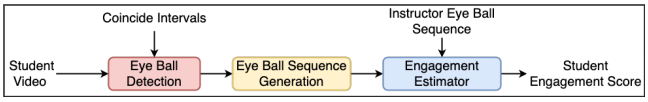

Figure 9: Student Engagement Estimation Framework

4.2 Student Engagement Estimation

The goal of the student engagement estimation module is to find out whether the student is following the presentation session. We assume that the *good presenter* looking at the screen is analogous to the following presentation. Therefore, we compare the eye movement sequence of the instructor and the student during the frames in the anchor-gaze time map. Finally, the percentage of the similarity is used to define the engagement score of the student for that course of time. Figure 9 shows the overall framework of the student engagement estimation module.

4.2.1 Eye Ball Detection. For comparing the instructor and the student's presentation followings, our first job is to detect the face followed by the eyeball. Therefore, we pick out the student and the instructor's video portion only for the frames belonging to the anchor-gaze time map. Then, we apply the Dlib 68 facial landmark model on each grayscaled frame to detect the person's eye landmarks. Each of the left and right eyes comprises 6 landmark points, and an eyeball exists inside that region. Thus, we pick up all the pixels within that region, considering the entire eye region. Each eye consists of two pixel-wise different parts, the cornea and eyeball. However, the distinguishable pixel is not fixed due to the ambiance of light reflection on the face. Therefore, we propose a clustering-based segmentation scheme where the cluster size is two. Specifically, we apply k-means clustering on all the grayscaled eye pixel values. The resultant clusters contain dark and light pixel values, respectively. The cluster with dark pixels is considered as eyeball pixels. We determine the eye center of the current frame by taking the average of all the points belonging to the eyeball cluster. Finally, the generated eyeball information is recorded as $< frame, eye_{index}, c_x, c_y >$, where, $frame$ denotes the frame number, eye_{index} is either of left or right eye, c_x, and c_y are the x and y axes values of the eyeball center, respectively.

4.2.2 Eye Ball Sequence Generation. In the current component, we generate the eyeball sequence during the anchor-gaze time map following the gaze gesture tracking method [6]. The eyeball center corresponds to the first frame of the anchor-gaze time map refer as a reference eyeball position. In the subsequent frames, we compute the magnitude shift by taking the difference of the eyeball center of the current and the next available consecutive frame. The magnitude shift is calculated according to Table 1. The reported eyeball sequence contains the structure $< frame, eye_{index}, symbol >$ where, *symbol* is the gaze shift symbol from Table 1.

4.2.3 Engagement Estimator. In the final phase, we match the generated eyeball sequences of the instructor and the student. Our intuition is the engaged student must follow the same eyeball sequence as the instructor. However, in reality, a marginal change in the eyeball center makes a different sequence outcome. Moreover, the sequence length is not fixed. Therefore, we figure out the eyeball sequence similarity using the longest common subsequence method for each set of sequences from the instructor and the student. Finally, the student engagement score is computed by taking the percentage of the longest subsequence length out of the instructor's anchor event frame count. In the presence of multiple anchor points in a single slide event, we simply average the engagement score. If the engagement score is above the threshold value ϕ, we consider the student is *engaged*. Otherwise, we mark the student as *non-engaged* for that slide.

Table 1: Magnitude shifts and corresponding symbols

x-axis shift	y-axis shift	Shift direction	Shift symbol
= 0	= 0	No shift	X
>0	= 0	Left	L
<0	= 0	Right	R
= 0	>0	Top	T
= 0	<0	Bottom	B
>0	>0	Top Left	M
>0	<0	Bottom Left	N
<0	>0	Top Right	O
<0	<0	Bottom Right	P

5 EXPERIMENTS AND OBSERVATIONS

For understanding the effectiveness of the proposed system, we have considered two types of real-time online lecture sessions – (S1) Online classroom teaching scenario and (S2) Group meeting with a formal presentation by a single presenter. We have investigated in total 4 lecture sessions, two from each of the types. Each online classroom lecture session lasts for one hour, whereas that value for each group meeting is 1 hour 20 minutes. For the classroom scenario, although we have 55 students in the class, only 6 are ready to share their videos. Therefore, the effective participants are 7, including the instructor. For the group meeting scenario, we have an average of 10 participants (min = 8, max= 12). Out of all participants, 2/3 are male, and the rest are female; 1/2 of the participants wear glasses. All the participants belong to the age group of 24 – 35 years. 90% of the participants are undergrads or research scholars, and the remaining are faculty members at the academic institution.

Ground truth annotation is one of the major challenging tasks for our system evaluation as engagement measure is subjective. For ground truth annotation, we have asked all the participants (the instructors and the students) to capture the front device screen using OBS studio. The participants use the laptops only with CPU computing units and a standard embedded webcam. The screen capture and the participants' video feed are used to generate the ground truth information.

The instructors choose the presentation topics for the group meeting scenario, whereas the classroom presentations contain the subject content. We have instructed the instructor to use different presentation content such as animation, image, and highlighted text. Apart from that, the participants are informed about inserting

slide numbers in the slide. They are open to using any presentation template. However, each presentation consists of a single template. Except for the ground truth annotation collection setup, there is no additional instruction to present the lecture. All participants but the instructor can interrupt during the lecture.

We have marked the participants as *Engaged* or *Non-Engaged* during a video slice (average duration of = 2 minutes, min= 1minute, max= 5 minutes) depending on whether the participant is looking at the lecture presentation for that window. Each video slice contains a single slide transition. We have shared the video slices with the participants and asked them to self annotate their video slices. The annotated data reveals that in 71% cases, the participants are *Engaged* and the remaining cases are *Non-Engaged*.

We compute the participants' (participated as students) *Engagement* score on a scale of $[0 - 100]$ based on the similarity in the gaze movement of the instructor and the students. Then, we decide a threshold point for partitioning the score into *Engaged* and *Non-Engaged* cases using *K-Means* clustering. Our assumption is each student is a mixture of *Engaged* and *Non-Engaged* status during the lecture session. Finally, the computed *Engagement* status is compared with the annotated value for calculating the accuracy, negative predictive rate and true negative rate of the system.

Table 2 summarizes the overall performance of the system for *80 percentile* cases of all the 4 scenarios, where $C1$ and $C2$ refer to the classroom lecture scenarios; and $M1$ and $M2$ refer to the group meeting scenarios. Our system achieves up to 93%(9%) accuracy for the classroom scenarios, whereas that value is 70%(7%) for the group meeting scenarios. Overall performance of the system concludes at an accuracy of 71%(10%). As we have captured the lecture sessions without restricting the participants, we have received a few participants' videos feeds with either improper environment conditions or occluded faces. If the unavoidable condition persists throughout the video, we exclude the participates from the further evaluation. Otherwise, even if the inevitable situation is present partially, we have processed the participants. In such cases ($M1$), the overall system accuracy is reduced due to the unidentified faces. For the students' understandability from the instructor perspective, *Non-Engaged* students identification is more important than the *Engaged* one. Therefore, along with the engagement detection accuracy, we have studied *Negative Predictive Value* and *True Negative Rate*. *Negative Predictive Value* defines true negative events encountered out of all the detected negative events. In contrast, the *True Negative Rate* illustrates true negative events out of the actual negative events. A high *Negative Predictive Value* of 91%(6%) states that our system detects less number of false-negative events. Hence, the system filter out the *Non-Engaged* students more precisely, and sharing that information of the *Non-Engaged* students is helpful to the instructor for understanding the students.

We further analyze individual participants' performance of group meeting scenario $M1$ in Table 3. Individually, the proposed system achieves accuracy, negative predictive value, and true negative rate up to 85%, 92%, and 92%, respectively. Although we receive 12 participants' video feed, we exclude three participants as environmental issues impact the face detection. Hence, further processing for the engagement score is not performed. Table 3 shows that the engagement status are poorly identified for the participants $u03$ and $u13$. A deeper look at the individual data states that the environment light varies rapidly for the participant $u03$. Therefore, for those poor light conditions, the face of $u03$ is not detected for the *Engaged* situation, and the accuracy of that participant is reduced. However, the accurate negative predictive value shows that the system precisely detects the *Non-Engaged* scenario. On the other side, the participant $u13$ was mostly drowsy, but he is trying to follow the session. Therefore, the annotation is marked as *Engaged*. However, due to the drowsiness, the eye is not detected properly and detected as *Non-Engaged*. This leads to a drop in accuracy value for $u13$.

Table 2: Engagement Detection System Performance: Individual Scenarios (*80 percentile* cases)

Online Teaching Scenarios	Accuracy (Avg.%(s.d.%))	Negative Predictive Rate (Avg.%(s.d.%))	True Negative Rate (Avg.%(s.d.%))
C1	***93(9)***	***100(0)***	***92(12)***
C2	*58(12)*	***100(0)***	*47(20)*
M1	*62(13)*	*75(14)*	*77(14)*
M2	*70(7)*	*88(12)*	*73(10)*
Average	***71(10)***	***91(6)***	***72(14)***

Table 3: Engagement Detection System Performance: Individual Participants of Group Meeting Scenario $M1$

Participants	Accuracy in %	Negative Predictive Value in %	True Negative Rate in %
u00	*46*	*50*	*86*
u03	*33*	***100***	*20*
u04	*46*	*75*	*55*
u06	*69*	*78*	*78*
u10	***85***	*92*	***92***
u11	*62*	*67*	*89*
u12	*62*	*86*	*60*
u13	*38*	*50*	*12*

6 CONCLUSION

To the best of our knowledge, the proposed approach is the first of its kind that captures discrete anchor events to incorporate student engagement. Moreover, our method is one of the systems that simultaneously captures both the instructor and the student's performance during the online session. While it captures the students' engagement or attentiveness as its primary design goal, the platform also signifies the quality of instruction by finding out the instances when a majority of the students are inattentive. The designed prototype is currently tested on two types of real-time online lecture sessions, and we observe a satisfactory performance (71% accuracy) of the system.

However, the designed prototype of the proposed platform is still in a nascent stage. We need to focus more on its real-time performance and perform a thorough usability study in the wild to assess its acceptability among the targeted users. Video processing

is always a heavy task, and therefore, we believe that further optimization of the system is possible. We keep these detailed studies as the future work for this system.

REFERENCES

[1] Mohammad Arif Ul Alam, Nirmalya Roy, Aryya Gangopadhyay, and Elizabeth Galik. 2017. A smart segmentation technique towards improved infrequent non-speech gestural activity recognition model. *Pervasive and Mobile Computing* 34 (2017), 25–45.

[2] Sinem Aslan, Nese Alyuz, Cagri Tanriover, Sinem E Mete, Eda Okur, Sidney K D'Mello, and Asli Arslan Esme. 2019. Investigating the impact of a real-time, multimodal student engagement analytics technology in authentic classrooms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[3] Mihai Bâce, Sander Staal, and Andreas Bulling. 2020. Quantification of Users' Visual Attention During Everyday Mobile Device Interactions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[4] Samit Bhattacharya, Viral Bharat Shah, Krishna Kumar, and Ujjwal Biswas. 2021. A Real-time Interactive Visualizer for Large Classroom. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11, 1 (2021), 1–26.

[5] Emily Jensen, Meghan Dale, Patrick J Donnelly, Cathlyn Stone, Sean Kelly, Amanda Godley, and Sidney K D'Mello. 2020. Toward automated feedback on teacher discourse to enhance teacher learning. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–13.

[6] Pragma Kar, Samiran Chattopadhyay, and Sandip Chakraborty. 2020. Gestatten: Estimation of User's Attention in Mobile MOOCs From Eye Gaze and Gaze Gesture Tracking. *Proceedings of ACM on Human-Computer Interaction* 4, EICS (2020), 1–32.

[7] Rebecca Killick and Idris Eckley. 2014. changepoint: An R package for changepoint analysis. *Journal of statistical software* 58, 3 (2014), 1–19.

[8] Thomas W Price, Joseph Jay Williams, Jaemarie Solyst, and Samiha Marwan. 2020. Engaging Students with Instructor Solutions in Online Programming Homework. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–7.

[9] Oleg Špakov, Diederick Niehorster, Howell Istance, Kari-Jouko Räihä, and Harri Siirtola. 2019. Two-way gaze sharing in remote teaching. In *IFIP Conference on Human-Computer Interaction*. Springer, 242–251.

[10] Gahyun Sung, Tianyi Feng, and Bertrand Schneider. 2021. Learners Learn More and Instructors Track Better with Real-time Gaze Sharing. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–23.

[11] Laton Vermette, Joanna McGrenere, Colin Birge, Adam Kelly, and Parmit K Chilana. 2019. Freedom to personalize my digital classroom: Understanding teachers' practices and motivations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[12] Xiang Xiao and Jingtao Wang. 2017. Undertanding and detecting divided attention in mobile mooc learning. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2411–2415.

[13] Matin Yarmand, Jaemarie Solyst, Scott Klemmer, and Nadir Weibel. 2021. "It Feels Like I am Talking into a Void": Understanding Interaction Gaps in Synchronous Online Classrooms. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–9.

[14] Iman Yeckehzaare, Tirdad Barghi, and Paul Resnick. 2020. QMaps: Engaging Students in Voluntary Question Generation and Linking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–14.

[15] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2015. Appearance-based gaze estimation in the wild. In *Proceedings of IEEE conference on computer vision and pattern recognition*. 4511–4520.

[16] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence* 41, 1 (2017), 162–175.

[17] Zoran Zivkovic. 2004. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of* 17^{th} *IEEE International Conference on Pattern Recognition, 2004.*, Vol. 2. 28–31.