# Webcam-based Attention Tracking in Online Learning: A Feasibility Study

**Tarmo Robal**
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia
tarmo.robal@ttu.ee

**Yue Zhao, Christoph Lofi and Claudia Hauff**
Web Information Systems
Delft University of Technology
Delft, Netherlands
{y.zhao-1, c.lofi, c.hauff}@tudelft.nl

## ABSTRACT
A main weakness of the open online learning movement is retention: a small minority of learners (on average $5-10\%$, in extreme cases $< 1\%$) that start a so-called Massive Open Online Course (MOOC) complete it successfully. There are many reasons why learners are unsuccessful, among the most important ones is the lack of *self-regulation*: learners are often not able to self-regulate their learning behavior. Designing tools that provide learners with a greater awareness of their learning is vital to the future success of MOOC environments. Detecting learners' loss of focus during learning is particularly important, as this can allow us to intervene and return the learners' attention to the learning materials. One technological affordance to detect such loss of focus are webcams—ubiquitous pieces of hardware available in almost all laptops today. In recent years, researchers have begun to exploit eye tracking and gaze data generated from webcams as part of complex machine learning solutions to detect inattention or loss of focus. Those approaches however tend to have a high detection lag, can be inaccurate, and are complex to design and maintain. In contrast, in this paper, we explore the possibility of a simple alternative—the presence or absence of a face—to detect a loss of focus in the online learning setting. To this end, we evaluate the performance of three consumer and professional eye/face-tracking frameworks using a benchmark suite we designed specifically for this purpose: it contains a set of common xMOOC user activities and behaviours. The results of our study show that even this basic approach poses a significant challenge to current hardware and software-based tracking solutions.

### ACM Classification Keywords
H.5 Human-centered computing: Human computer interaction; Empirical studies in HCI

### Author Keywords
Eye tracking; Face detection; Online learning; MOOCs



*IUI'18,* March 7–11, 2018, Tokyo, Japan



DOI: https://doi.org/10.1145/3172944.3172987

## INTRODUCTION
MOOCs have gained a lot of popularity over the past years and are now being offered to millions of learners on various platforms such as `Coursera`, `Udacity` and `edX`, among others. A major motivation behind MOOCs is the provision of ubiquitous learning to learners of all walks of life across the globe. Yet, despite their popularity, MOOCs suffer from low levels of learner engagement and learner retention, as only a very small percentage of learners who start a course actually complete it successfully; on average $5-10\%$ of learners succeed, in extreme cases this metric can drop below $1\%$ [9].

One reason why learners fail to complete MOOCs can be found in the design of the platforms. They tend to be rather basic (as a large amount of effort goes towards platform maintenance and scalability) and thus the delivery of course content does not always follow the latest educational findings. This setup contributes to the lack of self-regulation (in planning, motivation, goal setting) learners tend to exhibit, especially those without a higher education background [6]. In the MOOC context, loss of focus (during video watching, quiz submissions, etc.) is a core challenge, as it can have disastrous effects on learning efficiency [22]. Therefore, technological interventions which can detect a learner's loss of attention and can subsequently guide the learner's focus back to the course content could be of great value.

In recent years, a number of works have investigated inattention prediction based on various signals, including heartrate data [28], EEG data [29], skin conductance and temperature [4] as well as computer mouse pressure data [27]. While insightful, none of these approaches can be applied at scale in an online learning environment in the near future and thus, most of the existing research on inattention detection relies on eye-tracking data, including [1, 2, 3, 13, 17, 21, 30]. Here, the eye-mind link [18] is exploited as the eye gaze usually correlates well with a person's focus.

A major issue of existing eye tracking based inattention detection approaches is the lack of *real-time* detection capabilities (thirty to sixty second delays are common) [30]. An additional point of concern in our setting are the privacy requirements of MOOC environments—to ensure a learner's privacy all necessary computations should be conducted within the learner's *browser environment* (the alternative approach of streaming a learner's webcam data to a high-performant server has se-

vere privacy implications, while requiring the installation of dedicated software packages hampers usability)[1].

In this paper, we explore a significantly simpler alternative approach towards detecting a loss of focus whilst learning in a MOOC environment: we use the departure of a user's face from the webcam's viewport as a proxy for learner inattention—a user whose face is not aimed at the screen is unlikely to pay attention to a video playing on it. It turns out, that even this deceptively simple detection task is challenging in a MOOC environment where we have to consider widely varying consumer-grade hardware and browser software. In this paper we conduct an extensive study involving two open-source browser-based software frameworks for gaze and face detection, `WebGazer.js` and `tracking.js`, as well as a third hardware-based solution (a Tobii eye tracker) to determine an upper performance bound. Both software-based frameworks can be integrated into current MOOC environments, and perform all their processing on the user's computer without the need for a server infrastructure or additional browser plugins. We benchmark the ability of the three frameworks to reliably detect a user's focus towards the screen content (using the presence/abscence of a face as proxy) across a variety of common MOOC user activities such as watching a MOOC video whilst leaning on one's hand, checking the weather report on a smart-phone or drinking coffee.

We address the following research questions in our work:

**RQ1** *Which activities—that lead to different face positions in front of the screen—are typical for MOOC learners?* To this end, we compile a benchmark suite of fifty typical MOOC learner activities, partitioned into activities that are indicative of (i) focus, (ii) certain loss of focus and (iii) likely loss of focus.

**RQ2** *How reliable can current software frameworks detect the presence or absence of a face under typical MOOC conditions?* We conduct an extensive lab study involving `tracking.js` and `WebGazer.js` as well as a professional eye tracker (our upper bound in terms of performance). A total of twenty study participants execute the benchmark suite of activities in a controlled environment.

We find that in our setup, `tracking.js` performs significantly better than `WebGazer.js`, achieving a median detection accuracy of 62% across all fifty tasks (for the most difficult task detection accuracy was 17%), with the professional hardware-based eye tracker achieving a median accuracy of 72.5% (the most difficult task resulted in 27% accuracy). The observed detection delay is below two seconds for `tracking.js`, making it a viable choice for webcam-based attention detection (using face detection as a proxy). At the same time, the reported accuracy numbers suggest that current software and hardware solutions still struggle to provide a consistently high detection quality across all tasks.

[1]In order to investigate to what extent current MOOC learners are technologically able and willing to allow the MOOC platform to access their webcam feed (a necessity to make our vision of webcam-based attention tracking a reality) we conducted a preliminary study: we developed a working privacy-aware webcam intervention (i.e. none of the webcam data leaves a learner's computer) and deployed it to more than 800 learners in an engineering MOOC offered by TU Delft on the edX platform; we found 78% of learners had the technological capabilities (hardware, including a webcam, of sufficient quality) to run the intervention and 31% enabled it as learning support technology. Among those learners not enabling the intervention, about a quarter had privacy concerns, indicating that our vision is realistic for a significant number of MOOC learners.

## RELATED WORK: ATTENTION LOSS IN LEARNING

Different data collection methods have been used to study the loss of attention or focus of students in traditional classrooms since the 1960s, such as the observation of inattention behaviors [8], the retention of course content [12], using direct probes in class [23, 10] and relying on self-reports from students [5]. A common belief was that learners' attention may decrease considerably after 10-15 minutes of the lecture, which was supported by [23]. However, Wilson and Korn [26] later challenged this claim and argued that more research is needed. In a recent study, Bunce et al. [5] asked learners to report their attention loss voluntarily during 9-12 minute course segments. Three buttons were placed in front of each learner, representing attention lapses of 1 minute or less, of 2-3 minutes and of 5 minutes or more. During the lectures, the learners were asked to report their loss of attention by pressing one of three buttons once they *noticed* their attention loss. This led Bunce et al. [5] to conclude that learners start losing their attention early on in the lecture and may cycle through several attention states within the 9-12 minute course segments.

In online learning environments, attention loss may be even more frequent. Risko et al. [19] used three one hour video-recorded lectures with different topics (psychology, economics, and classics) in their experiments. While watching the videos, participants were probed four times throughout each video. The attention-loss frequency among the participants was found to be 43%. Additionally, Risko et al. [19] found a significant negative correlation between test performance and loss of attention. Szpunar et al. [24] investigated the impact of interpolated tests on learners' loss of attention within online lectures. The study participants were asked to watch a 21-minute video lecture (4 segments with 5.5 minutes per segment) and report their loss of attention in response to random probes (one probe per segment). In their experiments, the loss of attention frequency was about 40%. Loh et al. [11] also employed probes to measure learners' loss of attention and found a positive correlation between media multitasking activity and learners' loss of attention (average frequency of 32%) whilst watching video lectures. Based on these considerably high loss of attention frequencies we conclude that reducing loss of attention in online learning is an important approach to improve learning outcomes.

In a typical in-class setting a teacher has the ability to detect and re-gain learner attention through various pedagogical approaches. This is not applicable in MOOC environments due to the nature of online learning. Various technological approaches have been explored to detect and record signals of user (in)attention in the past besides eye tracking, includ-

ing heart-rate tracking through mobile cameras [28], brain activities through EEG analysis [29], skin conductance and temperature [4], posture and body pressure sensing and pressure applied on a computer mouse [27]. As already implied, most of the existing research though focuses on either face or eye-gaze detection [1, 2, 3, 13, 17, 21, 30].

Inspired by the eye-mind link effect [18], a number of previous studies [2, 3, 13] focused on the automatic detection of learners' loss of attention by means of gaze data. In [2, 3], Bixler and D'Mello investigated the detection of learners' loss of attention during computerized reading. To generate the ground truth, the study participants were asked to manually report their loss of attention when an auditory probe (i.e. a beep) was triggered. Based on those reports, the loss of attention frequency ranged from 24.3% to 30.1%. During the experiment, gaze data was collected using a dedicated eye tracker. In [13], Mills et al. asked the study participants to watch a 32 minute, non-educational movie and self-report their loss of attention. In order to detect loss of attention automatically, statistical features and the relationship between gaze and video content were considered. In contrast to [2, 3], the authors mainly focused on the relationship between a participant's gaze and areas of interest, i.e. specific areas in the video a participant should be interested in. In [30], Zhao et al. presented a method for detecting inattention similar to the studies in [13], but adapted and optimized for the MOOC setting.

All mentioned approaches relying on the eye-mind link share two common issues: they are usually unable to provide real-time feedback as they are trained on eye-gaze recordings with sparse manually provided labels (e.g., most approaches have a label frequency of 30-60 seconds, which directly translates into a detection delay of similar length), and the reported accuracy is too low for practical application (e.g., [30] reports detection accuracies of 14%-35% depending on training and video). As a result, we choose a different approach as discussed in the following sections.

## EYE AND FACE TRACKING FRAMEWORKS

Recall that we employ face presence and absence as proxies of learner attention and inattention respectively. Next to explicit face tracking software frameworks, eye-tracking frameworks are suitable for our work as well, as in the absence of a face, no eye tracking is possible.

In order to determine an upper performance bound, we use the professional high-end hardware eye tracker Tobii X2-30 Compact[2]. `Tobii` uses its own proprietary analytic software Tobii Studio to analyze the gathered eye tracking data.

Although there exist a number of different eye and face tracking software solutions, our choice is limited by the typical MOOC environment (which runs within the browser, and thus we require browser-based software frameworks), privacy aspects (all computations have to be performed on the user's device) and the variety of hardware capabilities we can expect MOOC learners' devices to have (the computations should not require too many resources). Evidently, JavaScript-based solutions fit the task description. Libraries such as `CCV.js`, `clmtrackr`, `headtrackr`, `ObjectDetect`, `tracking.js`, and `WebGazer.js` are thus potential candidates. After an initial testing phase of all mentioned frameworks we settled on two suitable ones: `WebGazer.js` [15][3] and `tracking.js`[4].

### WebGazer.js

`WebGazer.js` is an open-source eye tracking library written in JavaScript that is able to infer eye-gaze locations in real-time. Use-case specific extensions, e.g. to track users' web search behaviour [16] exist as well. `WebGazer.js` can be configured with different components to track gaze, pupils, or faces. We used the `clmtrackr` component[5], a face fitting library (referred to as `CLM` in the following), which has previously been used among others in works on camera-based emotion detection [20], and intelligent public displays in city environments [14]. `CLM` tracks a face and the coordinate positions of a face model, as shown in Figure 1. Using this face model, Webgazer can extrapolate the user's gaze (i.e., the point of the screen on which a user's gaze focuses) by estimating the face's distance and orientation from the screen. A weakness of `CLM` is its "agressive" face-fitting algorithm that often attempts to fit a model even when no face is present. This leads to many potential problems where random background elements (like posters, plants, furniture) are mistaken for faces, and sometimes even preferred over a real user's face clearly visible in the camera's viewport.

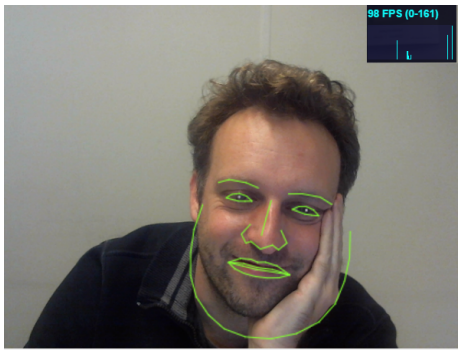


**Figure 1. Face fitting model generated by `CLM`. This example shows a common face fitting error due to hand positioning.**

### tracking.js

`tracking.js` is a JavaScript-based face tracking library (`TJS` in the following), which has been employed, among others, in security systems for identity verification [7] and object recognition tasks [25]. With respect to eye and face tracking, this library offers a significantly less powerful feature set than both `Tobii` and `CLM`, as it can only detect the presence and location of the boundary box of an object—in our case the face—in a video stream (see Figure 2). While it can also be employed to track the eyes' locations (but not the gaze), we do not use that feature in this study. We hypothesize that the

[2] `https://www.tobiipro.com/product-listing/tobii-pro-x2-30/`

[3] `https://webgazer.cs.brown.edu`

[4] `https://trackingjs.com`

[5] `https://github.com/auduno/clmtrackr`

simplicity of TJS leads to more reliable face presence and face absence detection.

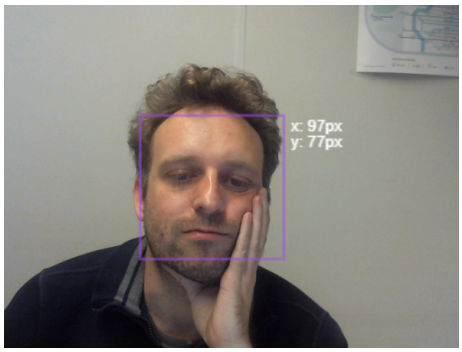


**Figure 2. Face Boundary Box detected by tracking.js**

### Detecting Face-miss events

We define a face-miss event to be an event of a user's face turning or moving away from the computer screen. The differences in the three evaluated frameworks (Tobii, CLM, TJS) leads to different heuristics for detecting a face-miss:

- Tobii: A face-miss event is detected if the proprietary Tobii Studio software cannot determine gaze point coordinates. This usually represents a problem with detecting the users' eyes by the tracker hardware (e.g., they are not within the camera viewport, they are closed, or obstructed by an object). At times, while the eyes can be found by Tobii, no gaze coordinates can be determined as the gaze direction is unclear. We cannot distinguish this case from a case where there is no face at all. In our experience, the presence of gaze coordinates is a very reliable proxy for the presence of a face (low false positive rate), while the lack of coordinates does not necessarily imply the absence of a face.
- CLM: Similar to Tobii, we define a face-miss event as the software's inability to fix exact gaze coordinates, which also means a failure in reliable face detection in case of CLM. In contrast to Tobii, due to the aggressiveness of the face fitting algorithm, CLM is is quite prone to detect faces where in reality there are none present (high false positive rate).
- TJS: We define a face-miss event as the library's inability to fix a face boundary box in the webcam's video stream. We do not try to track eyes or gaze.

The video or eye tracker stream is continuously processed while it is recorded. The Tobii system relies on dedicated hardware support for this task (which partially contributes to its high retail price), and is thus able to guarantee a sampling rate of 30 samples per second mostly independent of the computer hardware. For the webcam-based solutions, image processing of the video stream needs to be handled by the system's CPU and within the browser's environment. As a result, only low sampling rates are possible without overwhelming low-end computer systems. For this reason, we fix the sampling rate to 4 samples per second. However, due to the unreliability of the JavaScript timer events under high system loads, the standard deviation of the targeted sampling time of 250ms is 48ms in our experiments (described further in Section User Study). Furthermore, we have extreme cases where the sampling times increased up to 1157ms, i.e., less than one sample per second. Therefore, Tobii should be able to react with significantly lower delays than the webcam-based frameworks.

## USER STUDY

In order to evaluate the suitability of the chosen webcam toolkits for face and gaze tracking, we developed a **benchmark set of tasks**, which we argue represent common behaviours of online learners in front of their laptops. For each of the tasks we define the desired behaviour: the eye-tracking devices should either report the loss of the face/gaze (in the case of face-miss tasks) or keep detecting the face/gaze (in the case of face-hit tasks). We exclude mobile learners from these tasks as desktop learners are still the vast majority of learners in today's MOOC environments[6].

We designed a total of fifty tasks together with a small sample of regular MOOC learners (graduate students in our research lab). These tasks are—to some extent—abstract versions of the behaviour MOOC learners exhibit when watching lecture videos, one of the most common activities in so-called xMOOCs (i.e. MOOCs that are heavily relying on video lectures to convey knowledge, in contrast to cMOOCs which rely on learners' self-formed communities and peer teaching). The task descriptions we developed are shown in Table 3. They fall under three broad categories:

- face-miss tasks describe those user behaviours that *should* result in the loss of a detected face/gaze. Twenty-one tasks belong to this category; examples include *Take a sip from the cup [next to you] while turning away from the camera* or *Look straight up to the ceiling for 8 seconds*.
- likely-face-miss tasks should result in our frameworks reporting a mix of face hit and face miss samples. Two examples among the fourteen tasks in this category are *Lean back and put your hands behind your neck for 5 seconds* and *Draw a square on the paper*.
- Lastly, face-hit tasks describe user behaviours that should not influence our frameworks' ability to detect the face, though they may influence gaze detection. Fifteen tasks belong to this category, for example *Reposition yourself in the chair* and *Stare at the camera for 3 seconds*.

We developed a dedicated Web application as testing ground. The fifty tasks are presented as virtual "cue cards" to study participants and both TJS and CLM are included as webcam-based eye tracking solutions. The design of the application is modular, additional frameworks can easily be evaluated as well. We have open-sourced our application at **https://github.com/trx350/xMOOC_benchmark**.

The opening screen of the application is shown in Figure 3; an example task cue card is shown in Figure 4. The task order is randomized. The procedure for each task $Q_i$ is the same: the task description is shown and five seconds later a bell sound

[6]Concretely, based on a sample of twenty edX MOOCs offered at TU Delft, fewer than 20% of learners accessed the course content via mobile devices.

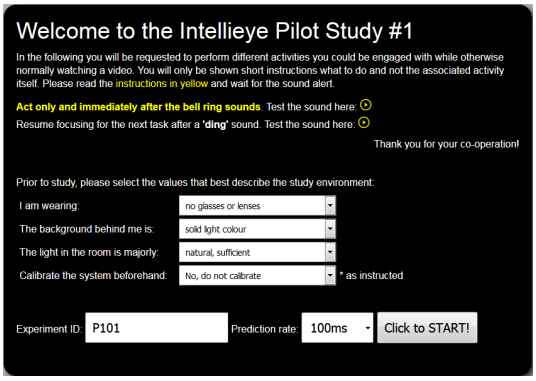


**Figure 3. Opening screen of the user study**

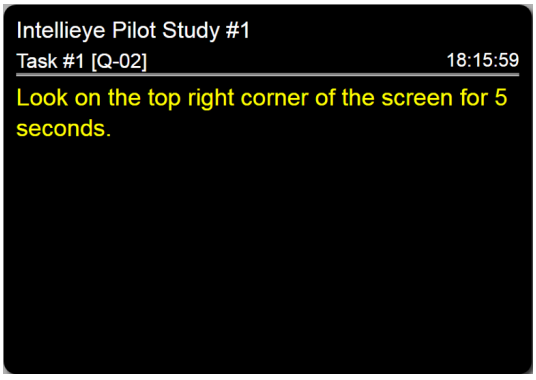


**Figure 4. Example task "cue card" of the user study.**

indicates the start of the task at time $t_{start}^{Q_i}$: at the sound of the bell the participant is expected to perform the task. Another bell sound (different to the one indicating the start) indicates to the participant when the task has been finished at time $t_{end}^{Q_i}$, and this is followed by the next task description. Task durations differ, depending on the specific task, e.g. Q31 requires a participant to look at a certain angle for 5 seconds while Q39 asks a participant to check his or her phone for 10 seconds.

### Study setup

We conducted all our experiments on a Dell Inspiron 5759 laptop (with built-in webcam situated in the center of the top screen bezel) with a 17-inch screen and a $1920 \times 1080$ resolution running Windows 10. The `Tobii` eye tracker was placed on the lower screen bezel.

The study was conducted across a one week period: twenty participants were recruited among TU Delft's graduate students and staff members via email lists. The participants did not receive compensation and spent less than an hour on this study. Among the twenty participants, nine wore glasses and two had contact lenses. In ten of the sessions the background behind the test subject had a uniform (light) color, in another 10 cases a poster or photographic background was observed. We recorded these settings in our study as we had conducted preliminary experiments which indicated that eye trackers (especially the software-based ones) can be mislead by noisy backgrounds.

As this is a controlled study, in order to facilitate the proper execution of the tasks, the participants were provided with the necessary tools to perform all necessary behaviours, including a sheet of paper and a pen (required for Q22, Q24 & Q25), a cup (Q41 & Q42) and a phone (Q39).

The `Tobii` requires a calibration step which participants concluded at the start of the study. The `CLM` framework can also be calibrated in a light-weight manner: five red dots are shown on the screen that have to be clicked one after the other. To test the effect of the calibration we randomly switched on the calibration step for eight of the twenty learners.

To prepare the participants for the tasks, each participant was trained on two tasks before the start of the actual study. The participants were reminded repeatedly to only start executing a task's required behaviour after the sound of the bell and to keep executing the behaviour until the ending sound occurred.

### Detection accuracy

For every task and participant, we determine the eye trackers' `face-hit/face-miss` predictions from the collected logs between the $t_{start}^{Q_i}$ and $t_{end}^{Q_i}$ timestamps. As the eye trackers vary in their sampling rate they all produce a differing amount of labels (face-hit, face-miss) for each sample interval. We evaluate the accuracy of the produced labels by computing the percentage of correct predictions (as defined by the type of task) in the task interval. For example, in a five second task slot the webcam-based approach takes a sample once every 250ms (on average), and thus we collect approximately 20 predictions. For a `face-miss` task, if 14 of the 20 predictions are a miss, the detection accuracy will be 70%. Lastly, we average the accuracy for each task across all participants.

**Table 1. `Tobii`'s delay between the start of a face-miss/likely-face-miss task and the first face-miss event. The data is averaged across all participants of a single task.**

| Delay | % of tasks |
|---|---|
| 1 sec | 53% |
| 2 sec | 28% |
| 3 sec | 6% |
| 4 sec | 3% |
| 5+ sec | 9% |

**Table 2. Overview of the impact of the participants' background on `TJS`'s and `Tobii`'s accuracy.**

| Background | # | Accuracy in % `TJS` | `Tobii` |
|---|---|---|---|
| Solid light | 10 | 61.5 | 68.6 |
| Poster/photo | 10 | 55.7 | 67.8 |

## RESULTS

In this section we report the outcomes of our user study along three dimensions: (i) accuracy across tasks, (ii) reaction times and (iii) the influence of the participants' background on the accuracy levels.

### Accuracy

The first question we consider is the accuracy of the three eye trackers under investigation across the fifty tasks of our benchmark suite. Table 3 lists the detection accuracy for each task, aggregated across the twenty study participants. As expected, `Tobii` achieves the highest accuracy, with an average of 68.2% across all tasks. Among the two software solutions, `TJS` clearly outperforms `CLM`, achieving an average accuracy of 58.6% compared to `CLM`'s 35.4%. If we were only to focus on the tasks where face misses and likely face misses form the ground truth, `CLM`'s accuracy would drop to 9.6%. The reason for this poor performance is `CLM`'s approach to face and gaze detection: it will try to match anything in the video frame to a potential face area, a separate face detection phase is not performed. This also explains its high accuracies in the face hits tasks. Note that the calibration step performed by some of our participants for `CLM` did not result in a different outcome.

The comparison between `Tobii` and `TJS` shows a relatively small performance gap between the Webcam-based eye tracker and the high-end device. While `Tobii` outperforms `TJS` in 39 of the 50 tasks, in many instances the difference in accuracies is rather small. Using `Tobii` as a reference point, `TJS` is able to conform with 77.8% of `Tobii`'s detected labels.

Due to the clear performance differences between `TJS` and `CLM`, in further analyses we focus exclusively on `TJS` and its performance compared to `Tobii`.

### Reaction Times

As one of the potential reasons for `TJS`'s lag in performance compared to `Tobii` we investigated the reaction times of both users and frameworks. More specifically, we measured the delay between the *instructed* start time of the task (i.e., the timestamp $t^{Q_i}_{start}$) and the first time a framework detects a face-miss. This time delta of course consists of both the user delay (i.e., the time it took for the study participant to finally start performing the task, which for some tasks—e.g. Q23 & Q46—showed a considerable delay) and the actual detection delay imposed by the framework. We averaged the delays of all participants for a task and report the percentage of tasks whose average delay is up to 1 second, up to 2 seconds, etc. in Table 1. For the majority of tasks, `Tobii` is able to detect the first face-miss within a second of the start of the task.

The `Tobii` eye tracker runs with a very high fixed sampling rate of 30 samples per second, and is mostly unaffected by the current CPU load of the host machine. Therefore, we make the assumption that the delays in Table 1 represent the user delay. In contrast, `TJS` and `CLM` can have very low sampling rates depending on the current system load (we aim at 4 samples per second, but we also experienced significantly lower rates). By comparing the times of detecting the first face-miss of both `TJS` and `CLM` with `Tobii`, we can obtain an intuition of the delays imposed by those frameworks. For `TJS`, this resulted in a delay of $0.6 \pm 1.1$ seconds, and for `CLM` in $1.3 \pm 1.0$ seconds. While these detection delays are not instantaneous, the delays are short enough for practical applications.

### Background as an Influencing Factor

As we conducted the user study in different rooms on different times of the day, we also recorded our participants with various backgrounds. In Table 2 we partitioned our participants according to the background they sat in front of during the study. All participants reported their background to be either of a solid light color (as present in many offices) or contain a poster and/or photo. This factor had an impact on the eye trackers' accuracy: while `Tobii`'s accuracy remained unaffected by the background, the `TJS` eye tracker considerably degraded when the background was noisy.

## SUMMARY AND DISCUSSION

In this paper, we have introduced the presence or absence of a face in a learner's webcam viewport as a simple proxy of learner attention or inattention in order to enable real-time attention tracking in a standard MOOC environment. This in turn will stimulate and support self-regulated learning.

We compared three potential technical solutions for this task: using the high-end professional Tobii X2-30 Compact hardware eye tracker, and using two software-based solutions that analyze the video stream of a consumer-grade webcam. We conducted a lab study with twenty participants, who had to perform a controlled benchmark suite of fifty realistic tasks, which introduced several challenging factors such as body movement, partially covering the face, noisy backgrounds, and crooked body postures. This benchmark suite and the accompanying Web application allows for a standardized and fair comparison of different approaches for `face-hit` and `face-miss` detection, and we provide it under an open-source license to foster future research.

Our experiments showed that the professional dedicated hardware solution outperforms the open-source software-based solutions both in respect to detection performance and processing speed, but is of course unsuitable for a large-scale deployment outside of a controlled lab setting. For the software-based solutions which can indeed run on typical hardware used by MOOC learners, the complicated `CLM` gaze tracking as employed by `WebGazer.js` introduces many complications, resulting in poor detection performance both for the presence and absence of a user's face. In contrast, the face tracking library `TJS` shows significantly higher performance for nearly all benchmark tasks. Additionally, both software libraries incur an additional time delay of around 1-2 seconds over the nearly instantaneous detection response of the hardware solution. With careful design, this delay should be easily manageable in a future MOOC learner attention detection component.

In our future work, we plan an implementation of an attention tracker suitable for a large-scale MOOC deployment on the basis of the `TJS` framework. Beyond purely technical or methodical challenges, this allows us to tackle additional interesting research questions: Would MOOC learners be willing to accept and use such an attention detection tool? What are the reasons why they would like/or refuse to use such technology? And of course finally, if learners accept the use of such tools, does this indeed positively impact their learning outcomes?

**Acknowledgments**
This research has been co-financed by the EU Widening Twinning project TUTORIAL and the Leiden-Delft-Erasmus Centre for Education and Learning.

**REFERENCES**

1. Stylianos Asteriadis, Kostas Karpouzis, and Stefanos Kollias. 2014. Visual Focus of Attention in Non-calibrated Environments using Gaze Estimation. *International Journal of Computer Vision* 107, 3 (2014), 293–316.
2. Robert Bixler and Sidney D'Mello. 2014. Toward fully automated person-independent detection of mind wandering. In *UMAP '14*. 37–48.
3. Robert Bixler and Sidney D'Mello. 2016. Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Modeling and User-Adapted Interaction* 26, 1 (2016), 33–68.
4. Nathaniel Blanchard, Robert Bixler, Tera Joyce, and Sidney D'Mello. 2014. *Automated Physiological-Based Detection of Mind Wandering during Learning*. Springer International Publishing, Cham, 55–60.
5. Diane M Bunce, Elizabeth A Flens, and Kelly Y Neiles. 2010. How long can students pay attention in class? A study of student attention decline using clickers. *Journal of Chemical Education* 87, 12 (2010), 1438–1443.
6. Dan Davis, Ioana Jivet, René F. Kizilcec, Guanliang Chen, Claudia Hauff, and Geert-Jan Houben. 2017. Follow the Successful Crowd: Raising MOOC Completion Rates Through Social Comparison at Scale. In *LAK '17*. 454–463.
7. Barbara Hauer. 2016. Continuous Supervision: A Novel Concept for Enhancing Data Leakage Prevention. In *European Conference on Cyber Warfare and Security*. Academic Conferences International Limited, 342–349.
8. Alex H Johnstone and Frederick Percival. 1976. Attention breaks in lectures. *Education in Chemistry* 13, 2 (1976), 49–50.
9. Katy Jordan. 2014. Initial trends in enrolment and completion of massive open online courses. *The International Review of Research in Open and Distributed Learning* 15, 1 (2014).
10. Sophie I Lindquist and John P McLean. 2011. Daydreaming and its correlates in an educational environment. *Learning and Individual Differences* 21, 2 (2011), 158–167.
11. Kep Kee Loh, Benjamin Zhi Hui Tan, and Stephen Wee Hun Lim. 2016. Media multitasking predicts video-recorded lecture learning performance through mind wandering tendencies. *Computers in Human Behavior* 63 (2016), 943–947.
12. John McLeish. 1968. *The lecture method*. Cambridge Institute of Education.
13. Caitlin Mills, Robert Bixler, Xinyi Wang, and Sidney K D'Mello. 2016. Automatic Gaze-Based Detection of Mind Wandering during Narrative Film Comprehension. In *EDM '16*. 30–37.
14. Masaki Ogawa, Takuro Yonezawa, Jin Nakazawa, and Hideyuki Tokuda. 2015. Exploring user model of the city by using interactive public display application. In *UbiComp/ISWC '15 Adjunct*. 1595–1598.
15. Alexandra Papoutsaki, Nediyana Daskalova, Patsorn Sangkloy, Jeff Huang, James Laskey, and James Hays. 2016. WebGazer: scalable webcam eye tracking using user interactions. In *IJCAI '16*. 3839–3845.
16. Alexandra Papoutsaki, James Laskey, and Jeff Huang. 2017. SearchGazer: Webcam Eye Tracking for Remote Studies of Web Search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. ACM, 17–26.
17. Matthieu Perreira Da Silva, Vincent Courboulay, Armelle Prigent, and Pascal Estraillier. 2008. Real-Time Face Tracking for Attention Aware Adaptive Games. In *ICVS '08*. 99–108.
18. Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372–422.
19. Evan F Risko, Nicola Anderson, Amara Sarwal, Megan Engelhardt, and Alan Kingstone. 2012. Everyday attention: variation in mind wandering and memory in a lecture. *Applied Cognitive Psychology* 26, 2 (2012), 234–242.
20. Filipe Santos, Ana Almeida, Constantino Martins, Paulo Moura de Oliveira, and Ramiro Gonçalves. 2017. Hybrid Tourism Recommendation System Based on Functionality/Accessibility Levels. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 221–228.
21. Kshitij Sharma, Patrick Jermann, and Pierre Dillenbourg. 2014. How Students Learn using MOOCs: An Eye-tracking Insight. In *EMOOCs*. 147–154.
22. Jonathan Smallwood, Daniel J Fishman, and Jonathan W Schooler. 2007. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin & Review* 14, 2 (2007), 230–236.
23. John Stuart and RJD Rutherford. 1978. Medical student concentration during lectures. *The Lancet* 312, 8088 (1978), 514–516.
24. Karl K Szpunar, Novall Y Khan, and Daniel L Schacter. 2013. Interpolated memory tests reduce mind wandering and improve learning of online lectures. *Proceedings of the National Academy of Sciences* 110, 16 (2013), 6313–6317.
25. Sajjad Taheri, Alexander Veidenbaum, Alexandru Nicolau, and Mohammad R Haghighat. 2017. *OpenCV.js: Computer Vision Processing for the Web*. Technical Report. University of California, Irvine.

26. Karen Wilson and James H Korn. 2007. Attention during lectures: Beyond ten minutes. *Teaching of Psychology* 34, 2 (2007), 85–89.

27. Beverly Woolf, Winslow Burleson, Ivon Arroyo, Toby Dragon, David Cooper, and Rosalind Picard. 2009. Affect-aware tutors: recognising and responding to student affect. *International Journal of Learning Technology* 4, 3-4 (2009), 129–164.

28. Xiang Xiao and Jingtao Wang. 2017. Understanding and Detecting Divided Attention in Mobile MOOC Learning. In *CHI '17*. 2411–2415.

29. Thorsten O Zander, Christian Kothe, Sabine Jatzev, and Matti Gaertner. 2010. Enhancing human-computer interaction with input from active and passive brain-computer interfaces. In *Brain-computer interfaces*. 181–199.

30. Yue Zhao, Christoph Lofi, and Claudia Hauff. 2017. Scalable Mind-Wandering Detection for MOOCs: A Webcam-Based Approach. In *ECTEL '17*. 330–344.

**Table 3. Overview of all fifty benchmark tasks, and the accuracy (in %) of `CLM`, `TJS` and `Tobii` averaged across the 20 participants in our user study. For "(LIKELY) FACE MISS" tasks we report the percentage of detected face misses, i.e. the eye tracker flags frames as not containing a face. For "FACE HIT" tasks we report the percentage of detected face hits. A higher percentage indicate a better performance. The best performance per task is shown in bold. ‡ Note that for tasks Q1 and Q2 Tobii's camera was not covered and the detection reflects participants' hand moving through Tobii's camera viewport to cover the webcam on the top bezel of the experimental laptop; for task Q43 there was no gaze to detect for Tobii.**

| QID | Task | Accuracy in % `CLM` | `TJS` | `Tobii` |
|---|---|---|---|---|
| | **FACE MISS Tasks** | | | |
| Q1 | Cover the camera for 2 seconds | 12 | **45** | 7‡ |
| Q2 | Cover the camera for 5 seconds | 28 | **73** | 17‡ |
| Q3 | Cover your face with both hands for 5 seconds | 17 | 67 | **75** |
| Q4 | Look what is under your table (3 sec) | 3 | 64 | **81** |
| Q5 | Stand up for 5 seconds | 10 | 68 | **71** |
| Q20 | Tilt your head to the right for 3 seconds | 15 | **59** | 38 |
| Q21 | Check if there is a HDMI port on the laptop | 12 | 56 | **77** |
| Q26 | Look straight up to the ceiling for 8 seconds | 12 | 72 | **92** |
| Q27 | Tilt your head back for 5 seconds (face ceiling) | 10 | 68 | **84** |
| Q28 | Tilt your head back for 2 seconds (face ceiling) | 5 | 51 | **66** |
| Q29 | Look down for 3 seconds | 4 | 35 | **78** |
| Q32 | Look left for 2 seconds | 7 | 50 | **72** |
| Q33 | Look left for 8 seconds | 14 | 69 | **88** |
| Q35 | Look over your right shoulder | 13 | 50 | **72** |
| Q36 | Look right for 10 seconds | 13 | 77 | **90** |
| Q37 | Look right for 3 seconds | 14 | 64 | **79** |
| Q38 | Look right for 5 seconds | 7 | 63 | **83** |
| Q39 | Check your phone for 10 seconds | 7 | 42 | **89** |
| Q40 | Check your phone, return after the ding | 13 | 37 | **87** |
| Q42 | Take a sip from the cup while turning away from the camera, return after the ding | 5 | 40 | **51** |
| Q47 | Look up and return immediately | 8 | 49 | **68** |
| | **LIKELY FACE MISS Tasks** | | | |
| Q6 | Lean back and put your hands behind your neck for 5 seconds | 2 | **67** | 63 |
| Q7 | Lean closer to the screen and immediately back | 3 | 17 | **27** |
| Q13 | Rapidly lean back and forth until the ding sounds | 6 | 37 | **57** |
| Q18 | Tilt your body to the left and stay for 3 seconds | 13 | 50 | **57** |
| Q19 | Tilt your body to the right and return immediately | 6 | 41 | **55** |
| Q22 | Draw a square on the paper | 9 | 45 | **67** |
| Q23 | Write down 5 keys left from letter A, focus back to the screen only after the ding | 4 | 19 | **61** |
| Q24 | Write down a sentence about weather | 15 | 47 | **73** |
| Q25 | Write down *I love Intellieye!* | 10 | 45 | **78** |
| Q30 | Look half-left and return | 7 | 36 | **64** |
| Q31 | Look half-right and stay for about 5 seconds | 7 | 42 | **77** |
| Q41 | Face the camera and take a sip from the cup until you hear the ding | 8 | 30 | **35** |
| Q46 | Cover the left side of your face with left hand over cheek and eye | 8 | 38 | **43** |
| Q48 | Look around in the room to every direction | 10 | 63 | **82** |
| | **FACE HIT Tasks** | | | |
| Q8 | Open browser and navigate to www.weather.com. Return after the ding. (15sec) | 94 | **97** | 80 |
| Q9 | Open new browser tab and return to this after the ding | **95** | 89 | 87 |
| Q10 | Open some program window (e.g. My computer) on top of study window and return after the ding | **99** | 87 | 94 |
| Q11 | Feeling sleepy? Yawn and cover your mouth with a hand. (3 sec) | **94** | 66 | 64 |
| Q12 | Grab the tip of your nose for 3 seconds | **100** | 64 | 71 |
| Q14 | Reposition yourself in the chair | **98** | 77 | 61 |
| Q15 | Scratch the top of your head (or nape) for 3 seconds | **94** | 69 | 85 |
| Q16 | Scratch the lower part of your left leg for 2 seconds | **93** | 79 | 64 |
| Q17 | Slowly lean back and stay for about 2 seconds | **96** | 32 | 38 |
| Q34 | Look on the top right corner of your screen for 5 seconds | 95 | 86 | **96** |
| Q43 | Rest your eyes for 5 seconds (close them) | **95** | 84 | 14‡ |
| Q44 | Scratch your left cheek for 3 seconds | **95** | 74 | 89 |
| Q45 | Sit still and face the camera for 5 seconds | **94** | 87 | 90 |
| Q49 | Grab your ears with both of your hands for 3 seconds | **95** | 76 | 85 |
| Q50 | Stare at the camera for 3 seconds | **95** | 89 | 88 |