Karlsruhe Institute of Technology

INSTITUTE OF INFORMATION SYSTEMS AND MARKETING

User Adaptive Systems Seminar

—

Eye Tracking for Attention Detection

Seminar Thesis

Joris Wessels, Marcel Namyslo, Paul Ferlitz,
Anne Hoff, Fabian Weschenfelder

At the Department of Economics and Management
Institute of Information Systems and Marketing (IISM)
Information Systems I

Reviewer:	Prof. Dr. Alexander Maedche
Supervisor:	Julia Seitz

20.08.2022

Contents

List of Figures

1. Motivation

1.1 Our eyes

Our eyes are a powerful input medium for receiving and interacting with information from our daily life. We can perceive incoming signals through different kinds of senses, such as hearing, touching, smelling, tasting, and as mentioned seeing. In contrast to those senses and information receivers, our eyes are involved in nearly everything we do and never stop interacting with the environment. They indicate what we are interested in and their movements are affected by what is happening in real-time. Therefore, our eyes provide a significant amount of information content available which underlines the high potential of using them for eye-based adaptive systems.

Some examples for those systems can be found in the research for neuro-adaptive systems as a form of user-adaptive systems, where the interest in capturing presumably objective data directly from the human body has increased massively. Additionally, one of the major strengths of neurophysical tools (EKG, eyetracker etc.) compared to other methods of collecting data is that the subjects cannot consciously manipulate their responses since these are not readily subject to manipulate (Dimoka et al., 2012).

These mentioned eye gaze and movement data can be tracked and collected by eye-tracking systems. They refer to the process of tracking different types of data such as gaze, focus, fixations, duration, and also TTFF (Time-to-first-fix). In this way, adaptive systems can reach a wide range from supporting in terms of decision making, analytical monitoring, taking over tasks, or helping fulfill them.

In our information-rich world, this kind of support is needed since receiving and processing information is limited: due to the flood of incoming information one‘s perception is not capable of processing every information we get to see (Sweller, 1988). We select and filter details, based on personal characteristics such as mental state, interest, and experience. This selective filter is what we refer to as attention (Broadbent, 1958). We are concentrating on a discrete aspect of information while ignoring other perceivable information. In addition to this, studies have shown that the amount of time a person can spend their attention on a certain thing without becoming distracted, has massively decreased in recent years (Shank (2017)). Therefore, supporting users in managing their limited attentional can bring up many new ideas for developing an eye-based user-adaptive system. In this seminar paper, we set our focus for our own adaptive system on the user´s attention in a very specific use case: during a usual online video meeting.

First, it is necessary to find out how attention or inattention could be detected during an online meeting. A person’s focus of attention can be identified in certain circumstances. Participants in a meeting, for example, might look at the speaker while they are listening to the talk or watch the content shared on the screen (Stiefelhagen et al., 2002) On the other side, he might be inattentive if he only looks at himself, the other participants, or out of his screen. He might not look at the screen and be drawing a little scribble on a piece of paper or be paying attention while taking notes on his sheet. As you can already imagine, detecting attention through gaze data might also involve some struggles which we will mention later in our work. But why could tracking attention during video meetings be so useful for a user?

1.2 Online Meetings

The pandemic and the resulting stay-at-home orders have led to significant changes in the way people work. One of these changes involves the increased use of video conferencing as a means of communicating or holding work meetings. Zoom, for instance, had 10 million daily meeting participants in December 2019, but by April 2020, that number had risen to over 300 million (Evans (2020)) students, pupils, and many ordinary workers. Also, surveys have shown that synchronous online meetings were associated with decreased engagement and attention by learners/workers (Weber and Ahn (2021)). While the amount of online meetings has increased on such a huge scale, it became very difficult for people to stay focused and attended during these meetings.

1.3 The research topic

We therefore decided to develop an eye-based, user-adaptive system with which we want to minimize this attention problem. In order to support users in properly allocating the limited attentional resources, we aim to develop and preliminary test an eye-based adaptive system that notifies users in states of inattention. By that we aim to answer the following research question: “How to design an eye base adaptive system that supports users’ attention in online meetings”

In order to track the gaze for means of detecting where the participant is looking at and then evaluating his attention, we tried to create a prototype that is built up on the data collected by the Tobii pro 5 eye-tracker. Due to the limitation of the provided resources by the eye tracker, we developed a program that can detect and indicate inattention as well as analyze and monitor the collected data for the user just by using these collected gaze data.

The application was then able to detect inattention during a meeting and to alert the viewer by small messages popping up. These pop-ups are meant to get the listener out of his inattentiveness and remind him that he needs to pay attention again. At the end of the meeting, the listener can see various statistics that analyze his attention as well as gaze data and display them in different ways. These statistics also include an individual calculated attention score, which gives the listener a positive feeling of success or a negative feeling if they were not paying attention at all, so they will try harder to pay attention next time. At the End of our project, we made a user study where participants could try out our prototype and give us feedback and their opinion about the functionality and effectiveness of our tool. In the ongoing, the functionality of our tool is described in chapter 2, chapter 3 summarizes the technological components. Next, in chapter 4 we highlight the conducted user study including the results and briefly demonstrate further improvements and an outlook und conclusion in chapter 5.

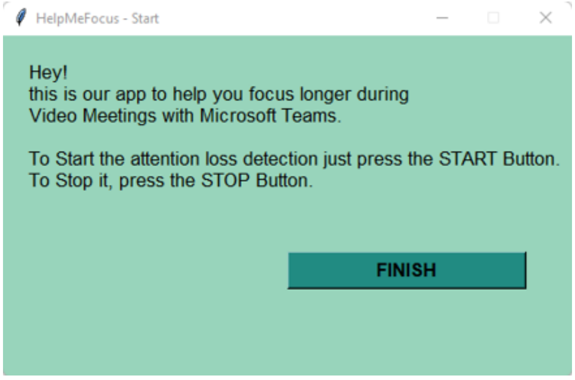
2. Functionality

The application consists of three main functionalities. Redirecting attention, learning from attention loss and preventing attention loss in the first place. The application implements these functionalities through three different graphical user interfaces. After explaining the general concept of the programm there will be a closer look at each of the three functionalities and its respective graphical user interface.

2.1 General concept

As already mentioned in 1, the primary goal of the application is to reduce inattentiveness in online meetings. Due to time restrictions the decision has been made to focus on online lectures and presentations which use the screen sharing option in Microsoft Teams. The general concept derives directly from the chosen use case. As lectures usually have a fix beginning and end, the programm needs to be started and stopped manually at the beginning and end of the lecture/presentation. Therefore a graphical user interface for intuitive starting and stopping the application is needed. Throughout the lecture/presentation the eyetracked person may loose attention. In order for the person to regain attention, he or she needs to be send a reminder to redirect the attention. A popup window will be shown instead of sending a sound signal as it seemed more fitting when handling gaze data. When sending a popup it is also possible to communicate the reason for the attention loss through a text message. For the mentioned popup window a second graphical user interface is needed. Last but not least, when the lecture/presentation has ended, it is important to give the user an overview of his or her attention throughout the meeting. For showing the respective statistics a third graphical user interface is implemented.

2.2 Preventing attention loss



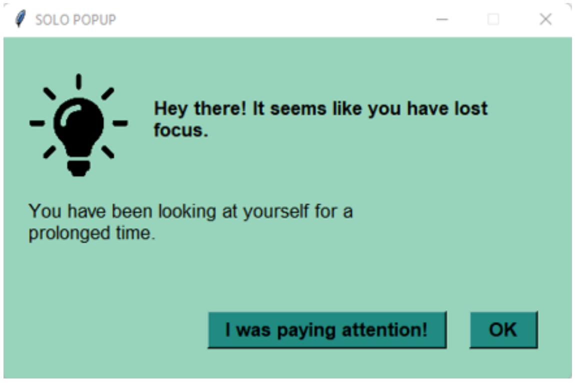
Hey!
this is our app to help you focus longer during
Video Meetings with Microsoft Teams.

To Start the attention loss detection just press the START Button.
To Stop it, press the STOP Button.

Figure 2.1: The Launch Interface after start of the recording

The first functionality is preventing attention loss from occuring in the first place. By starting the application and with that also starting the eyetracking, the user changes his or her behaviour and stays more attentive than without beeing recorded. This is quite similar to the Hawthorne Effect, which says that individuals who are beeing observed behave different from unobserved indivuals (Fox et al., 2008). Since in this case it is the user who explicity starts the recording, the Hawthorne Effect can have a huge impact on the general level of attention througout the meeting. The recording can be started trough the graphical user interface seen in Figure 2.1. The launching interface only presents the strictly necessary information. The user is heavily guided, as he/she does not have many options to interact with the programm. The main goal here is to not overload the user and drain his attention before even starting the meeting.

2.3 Redirecting attention



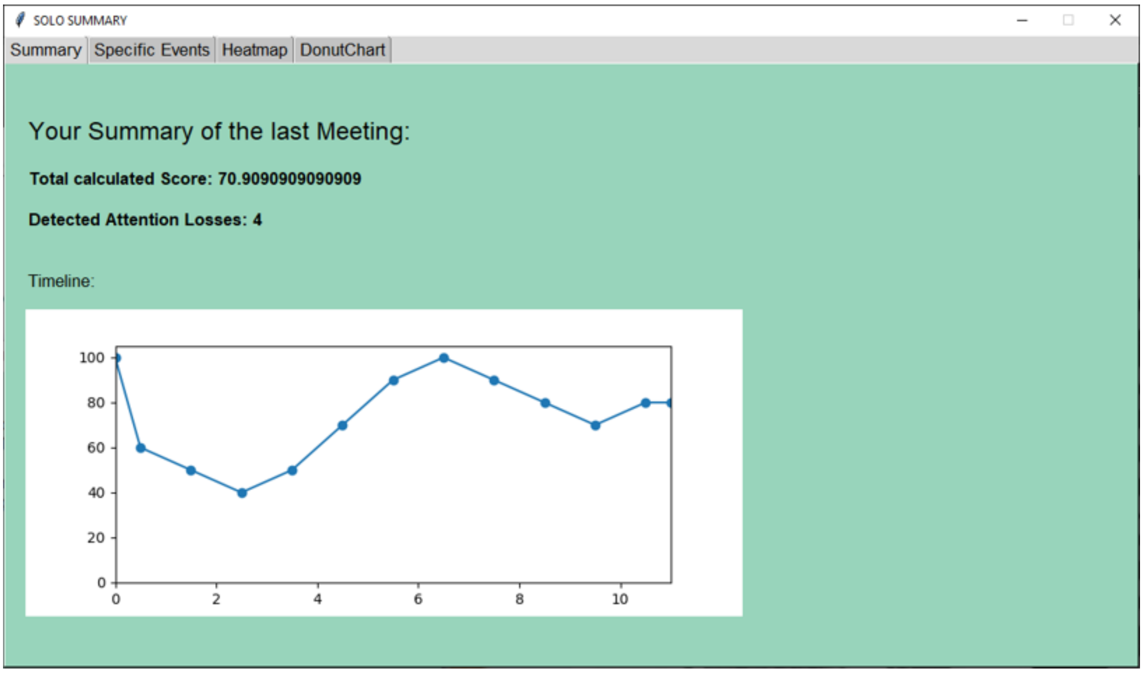
Hey there! It seems like you have lost focus.

You have been looking at yourself for a prolonged time.

Figure 2.2: Popup window for attention loss

In the optimal case, the following functionality would not be needed, as redirecting the users attention to the lecture only has to be done when an attention loss has occured. Since this is not an optimal world and simple external stimuli such as a police siren can draw the attention away from the lecture, this may be the most important functionality. Once an attention loss has occured, it is important to redirect the attention back to the lecture quickly in order to prevent the user from loosing pace with the presentation. If the user cannot keep up with the lecture the general attention will plummet as he/she is no longer able to participate in the presentation or understand the contents of the lecture. If the application has detected a loss of attention it will show a popup similar to the one in Figure 2.2. The application classifies the users attention as lost if the user has looked at a specific area for a prolonged period of time. Such areas include the leave button, the other speakers, the own camera or the non-screen area. The popup window is kept as minimalistic as possible. The goal is to redirect the users attention to the ongoing lecture and not to the application. Therefore the user should only notice the popup and then click it away to listen to the presentation. He/She should not have to read a lot of text or look at flashy pictures which may draw to much attention. The popup also shows the reason for the inital attention loss in a short sentence. Even though it is not strictly necessary information, by adapting the popup it can help the user to get rid of the thing drawing his or her attention. Has the user been looking at himself/herself a lot? It may be wise to turn off ones own camera.

2.4 Learning from attention losses



Your Summary of the last Meeting:

Total calculated Score: 70.9090909090909

Detected Attention Losses: 4

Timeline:

Figure 2.3: Popup window showing a summary of the meeting

The last functionality, the learning from attention loss, is enabled through the summary interface as seen in Figure 2.3. The summary interface shows comprised and aggregated statistics of the users attention throughout the meeting. The graph showing the attention development over time gives the user the ability to interpret his attention over time and adapt acordingly. A heatmap and a donutchart have also been implemented. The heatmap shows where the user has looked for how long and the donutchart gives an overview of how often an area has been looked at. Other statistics can easily be added to the currently available ones if needed. It is not only possible to improve the attention in a single meeting, but instead enable a learning process which may go on for several days or weeks and improve general atttention over the long term.

3. Implementation

This chapter will discuss which technologies and implementations were planned and lastly used in the final implementation.

3.1 Datasets

In the early stages of designing the application there was a discussion about using machine learning to aid the team's cause. For this, a dataset (or multiple) with sufficient source material would have been needed. But assumably for the reasons of privacy and the highly individual use case that the team is trying to solve, finding datasets with webcam footage or eyetracking data is nearly impossible. After searching different communities specialized for sharing and collecting datasets (e.g. reddit.com/r/datasets and paperswithcode.com), the team concluded to either collect the data itself (see section 3.2 Desktop and Web Solution), use public news report streams or rely on data given by the IISM chair. Because of the mentioned lack of datasets the team has agreed to not use machine learning for attention loss detection. Instead the team has opted to implement a rulebased system, making the use of a dataset unnecessary.

3.2 Propositions

Desktop Solution

The program would be fully implemented in Python using a local desktop approach. Ideally the program would be active at all times, hook into Zoom when a meeting is started and show results at the end when a meeting is over.

Desktop and Web Solution

The second option was to have a program as discussed in section 3.2 - Desktop Solution but also providing the team with a browser embedded Zoom client to take part in and additionally beeing able to record and label Zoom meetings more easily. This would be more versatile but also cost more time to develop.

3.3 Final Decision

For time and scope purposes the team decided to implement the pure Desktop Solution.

Firstly, the team agreed to use Microsoft Teams over Zoom, as the latter did not provide a consistent viewing experience (in browser and desktop application). Window sizes were not consistent, positions of different zones not configurable to a point where they were always the same and in general the overall stability of Zoom wasn't enough to guarantee a good synergy with the team's application.

Secondly, the team realized that the Tobii Interaction Library was implemented in and for the C++ programming language. Therefor C++ had to be added to the technology stack.

Lastly, a middleware had to be selected that would manage the data exchanged between the C++ and Python part of the application. Again for scope purposes a socket or similar IPC implementation was not used and instead all data was written to and read from a central CSV file. The final application processes as they have been implemented can be found in figure 3.1.

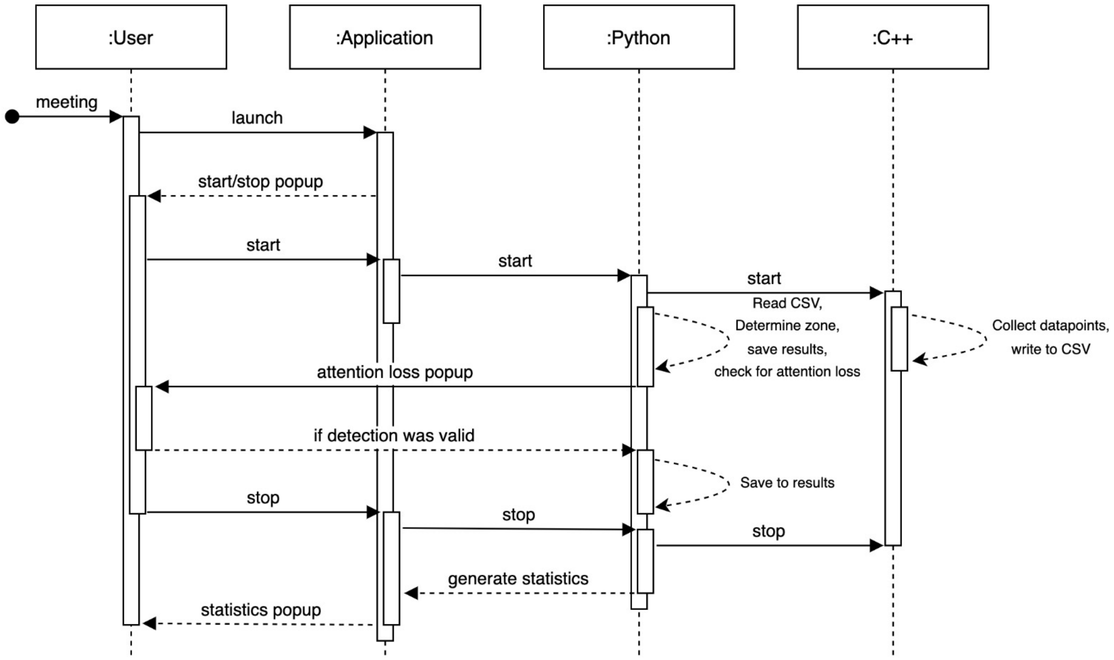


Figure 3.1: Diagramm Showing the Application Processes

3.4 Technology in use

Due to the scope of this seminar the team only implemented a MVP fulfilling all requirements and goals that were discussed during the weekly team meetings.

Non-default Python libraries in use:

- tkinter - for displaying graphical content
- pandas - to make working with large numbers and data arrays more easy
- Pillow - library that adds support for opening, manipulating, and saving images
- matplotlib - is a plotting library

Non-default C++ libraries in use:

- Tobii Interaction Library - for managing the eyetracker and processing the data generated by it

The program structure includes the following classes:

Python

- main.py - program entry point
- graphics.py - creates almost every menu seen during the execution of the application
- graphicsCalculations.py - provides the summary screen with an algorithm to generate the attention over time graph
- heatmap.py - functionality for generating a heatmap from the CSV data
- donutchart.py - functionality for generating a donutchart from the CSV data
- processing.py - manages the main application loop which also includes processing incoming CSV data and displaying the popups
- tools.py - provides the rest of the application with often needed methods, for example a confing reader and writer

All classes together result in an application that can read data provided in a CSV file, process it through the defined steps and algorythms to finally graphically output the results to the user.

C++

- main.cpp - program entry point which collects 60 datapoints per second and saves them to the dedicated CSV file

This singular class handles all interactions with the Tobii Eyetracker and saving the collected datapoints to the defined CSV file.

The above mentioned technologies were as already stated implemented to present a MVP. Allthough the overall complexity of the application is fairly minimal, it does demonstrate what can be acchieved with little input data as eye gaze X and Y coordinates. The general structure of collecting, saving and processing the data to then finally give a response to the user is only a minor insight into what can be done. More features and additions, as for example machine learning, can easily be implemented by other teams continuing work on this project in the future.

4. User Study and Results

The following sections describe the setup and procedure of the testing and presents the results as obtained from interviews.

4.1 Testing: Procedure and Sample

In the following, the procedure of the user study is described, and information regarding the sample is given.

4.1.1 Testing Procedure

The participants of the Explorative User Study came to IISM and the team made them sign the obilgatory agreement with respect to data protection and privacy. Then the configuration of the eye-tracker, or, respectively, its individual calibration was done. After the recording of the team's program was started, the study participant was to participate in an online video meeting in the course of which they were presented a video lecture of a duration of about ten minutes. This video lecture was a sample from the lecture "Rechnerorganisation" at KIT, held by Dr.-Ing. Lars Bauer and Prof. Dr.-Ing. Jörg Henkel. After finishing, the participant used the provided dashboard. The team took screenshots of their dashboard results. Finally, the head of the study conducted a semi-structured interview with respect to the participant‘s perception of the software prototype.

4.1.2 Interview Questionnaire

At the testing stage of this study, the recruited participants were asked questions regarding their personal user experience with the tool and their very subjective evaluation of the tool. The questions were subdivided into the sections General User Perception, Popups, Dashboard, and Visualizations.

Additionally, the opportunity for feedback beyond the prepared questions was given.

The team created the questionnaire in both German and an English translation. All recruited participants are German native speakers which is why the ultimately only deployed the questionnaire in its German version; irrespective of this, in the Appendix both versions (German and English translation) can be found and used for potential follow-up studies.

4.1.3 Sample

The sample of the Explorative User Study is a "Convenience Sample", i.e. being recruited from the team's social circles (friends, family, acquaintances) and consists of six participants; four identified themselves as male, two as female. They are between 21 and 31 years old. All of them are students, mainly at KIT.

4.2 Results: User Perception and Evaluation

The following chapter reports the users' perceptions and evaluations as obtained from the interview questionnaire (4.1.2); corresponding implications and interpretations are discussed and, if applicable, conclusions are drawn.
In the first sub-section (4.2.1), user perceptions are reported on an aggregated level, i.e. concerning those five questions that were to be rated on a scale from 1 (representing ”no/very low/...”) to 5 (representing ”yes/absolutely/strongly/...”).
The second sub-section (4.2.2) deals with the qualitative results, i.e. the user perceptions, verbally expressed as responses to open-ended questions in the semi-structured interviews, are reported. This section follows the structure of the questions in the questionnaire, i.e. it is subdivided into the sections General User Perception, Popups, Dashboard, and Visualizations.

4.2.1 Aggregated Results from User Ratings

Figure 4.1 depicts the summary of the results from those five questions that the users had to rate on a scale from 1 (representing ”no/very low/...”) to 5 (representing ”yes/absolutely/strongly/...”).
Overall, the team's program can cautiously be interpreted to have been fairly well perceived, as can be deduced the diagram reporting the ratings given by the six study participants.

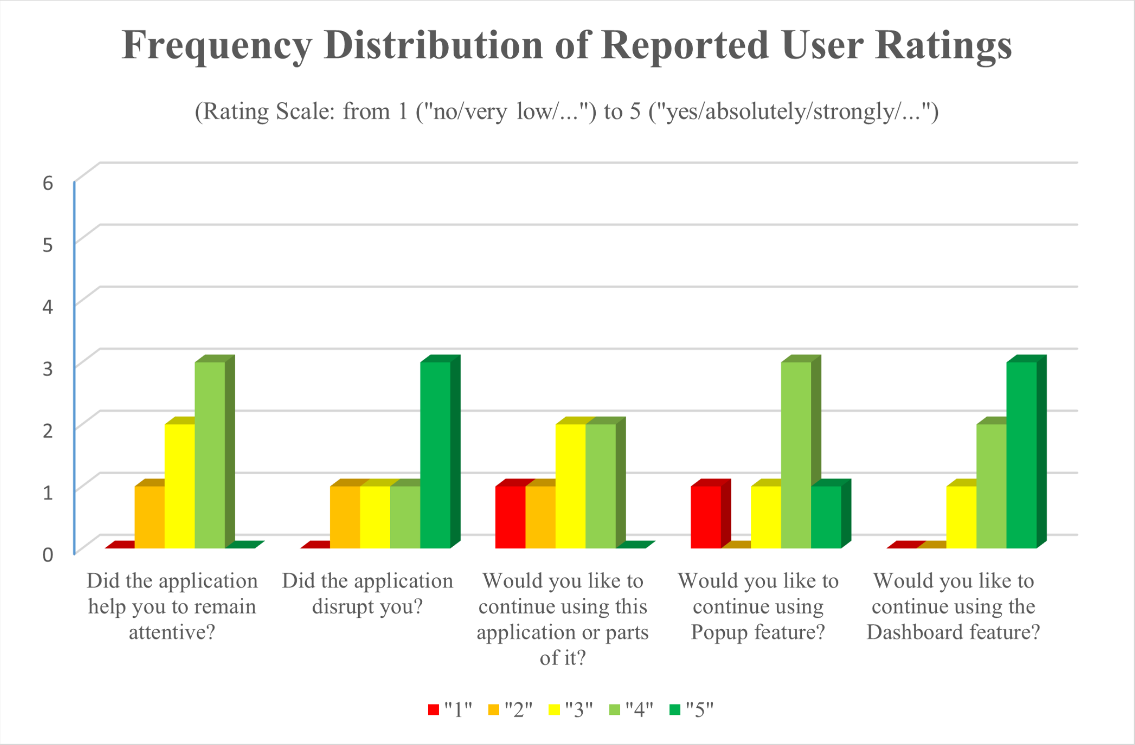


Figure 4.1: Frequency Distribution of the Reported User Ratings

Though, required sample size is the required number of experimental units to be included in a study in order to be able to answer the research questions (Noordzij et al. (2011)). Consequently, the team's urgently need to stress that, given our extremely small sample size of six participants, from a statistical perspective the team is very far from a sample size that would allow to draw any conclusions. Lakens (2022) provides a detailed overview of multiple approaches for the justification of the sample size in a study design depending on the overall goals of the collection of data, available resources, and deployed statistical analysis approaches (Lakens (2022)). In light of the explorative nature of this study, first, the team would like to dissociate ourselves from any attempt to perform statistical inference based on the data collected during the interviews, but second, the team still considers it of no harm to report results on an aggregated level as well as on the level of individual or anecdotal evidence, while keeping in mind the sample size.

The fact that for the question, how helpful the tool was considered for remaining attentive, the most common answer was a 4 out of 5 creates the strong impression that the way the team has set up our tool was reasonable and capable of achieving its goal. Regarding eventual disruptiveness, most participants reported to have not at all felt disrupted by the tool in any way. The final three rating questions addressed the question of a possible intended further use in the future. Regarding a possible further use of the application, or parts of it, in general, most frequent answers were 3 and 4 out of 5, which might reflect the summation of the perception of some of the flaws of the program, as will be discussed in the following sections. The Popup feature, when solely asked for, was assigned a higher rating (most common answer being a 4 out of 5). In comparison, when solely asked if they could image continuing using the Dashboard feature, participants most frequently responded with a 5 out of 5, leading to the conclusion that the Dashboard feature is a highly appreciated part of the program.

These results can serve as a first sense of the users' perception of our tool. Again, the team does not attempt to interpret the results in an inadmissible way – however, we consider them useful in conjunction with the qualitative results obtained from those interview questions that were to be openly answered and whose results are shown in the section successive to this one.

4.2.2 Qualitative Results from Study Interviews

This section deals with the verbally expressed perceptions and follows the structure of the questions in the questionnaire, i.e. it is subdivided into the sections General User Perception, Popups, Dashboard, and Visualizations.

4.2.2.1 General User Perception

Overall, the participants expressed to like the basic idea of the tool. It was perceived to be restrained in a positive way, and someone liked that *"[i]t immediately reminded [them] to pay attention to the screen again"*. Another positive feedback was that someone felt that merely for the reason of the program running in the background, they subconsciously automatically turned to be more attentive to the screen and, as a consequence, felt an overall increase in attention. Another person, whose self-assessment was reportedly to be

a comparably attentive person, expected the tool to be especially helpful in a noisy or restless environment.
On the negative side, users reported some technical flaws. Some people considered the tool to be too sensitive, while others considered it to be insufficiently sensitive. One person was not happy with the fact that when inattention has been detected, the tool's popup first draws the user attention to the program rather than directly to the main content. Someone felt a compulsion to constantly look at the screen and expressed fears that this could perhaps also be counterproductive over a longer period of time. Someone expressed doubts regarding real benefits of the tool's deployment in real-world scenarios. For some users' taste, explanations in the dashboard part of the tool were missing, and thereby sacrificing a reasonable comprehension. One person did not like the requirement of extra hardware (i.e. the eye tracker) for the usage. Furthermore, one person was seriously concerned about their privacy and expressed the feeling of surveillance and being monitored all the time.

4.2.2.2 User Perception: Popup

The User Interface of the popups was mostly appreciated (e.g. *"nice buttons"*, *"nice colors"*), despite for one person's taste it was *"too minimalistic"* or even *"ugly"*. For someone, the provided texts were too long, while most people considered them to be short and unambiguous and that they were easy to get quickly rid of after an inattention detection. Overall, the popups were appreciated as *"meaningful"*, *"undisturbing"*, *"effective"*.

4.2.2.3 User Perception: Dashboard

Some person said that it was *"[i]nteresting to see when you paid more attention and when less"*. Another person expressed the idea that *"[i] could support continuous improvement across individual sessions"*. Several users described them to be *"meaningful analyses"*. However, for some users *"[t]he display and listing of the different events was not particularly helpful"*. Two users were unhappy with the dashboard with respect to the perceived comprehensibility – they complained brief explanations to be missing and, consequently, to be unable of using the full potential of the dashboard information. In that context, someone explicitly mentioned the graph with the timeline.

4.2.2.4 User Perception: Visualizations

Regarding the visualizations, the user perception was divided: By some users, the Heat Map was considered to be intuitive, while others found it was confusing; some users found the Donut Chart to be relatively less intuitive, whereas others deemed it comparably more intuitive, especially *"with no previous knowledge"*. Corresponding to their perception of the dashboard, someone perceived the provided labels as not entirely clear. *"The Heat Map concept is nice, but I don't know if it's really helpful"*, was another person's experience; on the other hand, *"[in the donut chart] you can see well in the comparison of the total time, on which you have spent how much time"* was some user's appreciation. Another user expressed their experience in a nutshell as *"I would love to use both tools"*.

5. Conclusion and Further Improvements

This chapter will give a conclusion of the prototype and will explore some technical issues that occured, and how they could be dealt with in the future. In addition to that some explorative aspects will be discussed, as well as an implementation of a machine learning approach.

5.1 Conclusion

After the process of resarching, implementing, testing and evaluating, we ended up with an eye based user adaptive tool, which is able to detect inattention in given circumstances und help the user paying attention during an online meeting. It is able to assume inattention and to alert the viewer in an simple interactive way by only collecting the data in real time and showing plain and simple popups. This was also the main result of our user study, which resulted in an all in all positive feedback. However, there are still many aspects of improving our tool and problems which accured during the project, which will be explained in detail in the following few paragraphs.

5.2 Further Improvements

5.2.1 Technical Issues

During the implementation and testing of the application, a few technical issues prevented accurate detection.

5.2.1.1 Eyetracker Accuracy

An accurate measurement of the x and y coordinates of where the subject is looking is vital in order to get good results. If for example, the subject looks at an area of interest but the eye tracker wrongly places the point he is looking at wrongfully slightly out of the area, then the program will not trigger an event even though it should. Unfortunately, the eyetracker itself presented several problems concerning accuracy.

First, the calibration was sometimes inaccurate - especially around the edges. Specifically a lot of the areas of interest, like the ”leave button” or the upper and lower task bars, are located at the edge of the screen. Thus, an inaccurate calibration can lead to events that involve these areas not being triggered.

Another issue with the eye tracker that presented itself during testing was its limited range. If the subject moved too much back and forth or up and down, he would move out of the reach of the eyetracker. This would result in an incorrect detection or detection would fail completely.

As a countermeasure, the definitions of the zones could be adapted. If they were made bigger, there would be more room for slight inaccuracies by the eye tracker. On the other hand, this would increase the possibility of false positives.

5.2.1.2 Blinking

Another issue that presented itself was that while someone was blinking, the eye tracker couldn't determine the location of the eyes on the screen. This was evaluated as if the person was looking off the screen. The result was that every time the subject blinked, the timer that determined when an event was triggered was reset. This resulted in a large number of false negatives. Because a person blinks approximately 12 times per minute, and a blink lasts around 1/3s (Kwon et al. (2013)) - 20 times longer than the interval in which the eye tracker collects data - this was a very important issue. To solve this problem, the data points taken while blinking should be excluded. This can be done by excluding "out of bound" data points that last 1/3s or less.

5.2.2 Explorative Aspects

There are also some aspects that could be explored further in the future.
When deciding on how to detect the soll of attention two things were picked more or less at random.
One of them was the trigger time, so the time the subject has to look at a certain area in order for the program to trigger an event. The attempt was to find a balance between too many false positives because of a timeframe that is too short, and too many false negatives because of a timeframe that is too long. But being inattentive does not start at a specific timeframe. A subject can look very long at a specific point and still be very concentrated. On the other hand he can also look only briefly at something, but be very inattentive.
Another aspect that was determined based on assumptions, was which events were a sign of inattentiveness. The team drew from their own experiences with inattentiveness in meetings, but during testing it became clear, that these events didn't apply to everyone.

5.2.3 Machine Learning

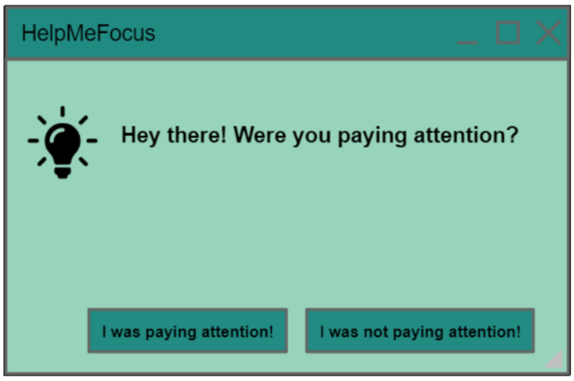
One option to deal with the above mentioned explorative aspects (5.2.2) would be to abandon the specific timeframe and events and let the detection be handled by a machine learning approach.

5.2.3.1 Data collection

As there is no appropriate labeled training data available, it would have to be created specifically for the application. As the application is also supposed to be as user adaptive as possible, it is advised to generate data for each user. For collecting such test data, Hutt et al. (2018) describe a method of detecting mind wandering by so-called probe-caught mind wandering.
Probe-caught mind wandering sends pseudo-random thought probes. These probes ask the user whether they have been attentive or mind wandering in the timeframe directly before the thought probe.
For the requirements of this application it is suggested to use a variation of this concept. In a setup meeting the user will be presented with a popup (see figure 5.1), where he is asked whether he was paying attention or not. His answer will then be matched with the

recorded data points before the probe appeared. This will create the labeled data that can then be used to train the machine learning model.

In addition, the training data can be expanded during regular execution of the program.



Hey there! Were you paying attention?

Figure 5.1: Mockup of a prompt for probe-caught attention detection

When the system detects loss of attention the user can either click the ”OK” button, or he can label the detection as being false by clicking ”I was paying attention”. This feedback can then be used to improve the model.

5.2.3.2 Machine Learning Models: Neural Networks (NN, RNN, LSTM)

From the field of basically applicable Machine Learning methods, we consider Neural Networks (”NN”) to be particularly suitable for our research purpose, since multilayer feedforward NNs are deemed to be capable of approximating any non-linear function which renders them ”universal approximators”, given an appropriate so-called activation function is selected (cf. e.g. Hornik (1991); Csáji et al. (2001)).

As discussed in e.g. Goodfellow et al. (2016), the goal of a basic feedforward NN is the approximation of some arbitrarily complicated function through a composition of simpler functions. The basic building block of such a feedforward NN is a simple node (”neuron”) which does a linear transformation of input data – by multiplying the input data by a number (”weight”) and adding a constant (”bias”) – and a subsequent application of a nonlinear, differentiable function (”activation function”). Many such neurons can be stacked as ”layers” of neurons to build a multilayer NN (while each layer may be composed of more than one neuron). Such a multilayer NN consists of three or more layers: the input layer, one or more intermediate (”hidden”) layers whose neurons are connected in a feedforward way to the following layer’s neurons with a weight parameter, and the output layer. The NN works by forward-propagation – reception of the input (”ground truth”) data by the input layer, computation of each of the (stacked) hidden-layer-neurons’ output on the basis of its weighted input and the activation function, up to the output layer. Through a loss function, the resulting error in the output is compared with the desired output (the ground truth) which was fed into the NN. Learning in a NN is achieved by backpropagating the error to the weight and bias parameters and then updating those parameters by taking the gradient with respect to the loss in such a way that the output error is reduced.

For better handling of ordered sequence data (e.g. time series data), Recurrent Neural Networks (”RNN”) (see e.g. Elman (1990), Rumelhart et al. (1986), Goodfellow et al. (2016)) have a recurrent hidden state as a memory for dependencies over time, which allows mapping an input sequence into an output sequence whose elements basically depend on all the previous time steps, RNNs learn recurrent functions via backpropagation through time algorithm.
Long Short-Term Memory Units (”LSTM”), proposed by Hochreiter and Schmidhuber (1997), comprise three gates regulating the information flow into as well as out of the memory cell: (i) an input gate, (ii) output gate and (iii) forget gate. The output gate controls the degree of information from the memory cell that is used additionally. The content of the memory cell is updated at each time step: the forget gate regulates the extent of the memory cell is retained or discarded (”forgotten”) and the input gate modulates the extent to which the new memory content is added to the memory cell. In contrast to RNNs whose simple recurrent units’ contents are overwritten at every time step, this self-control over the existing memory allows LSTMs the operation at multiple arbitrary time intervals of the sequential input data.

5.3 Final answer to research question

For developing a user-adaptive system that deals with collecting gaze data from a user and thus inferring the attention of the user, a clear definition of inattention and how it should be recognized is the most difficult and urgent question. As attention is just a concept that doesn‘t have its own significant parameter like a metric system, we developed up our own definition and rules of when a user is inattentive. This definition is based on our individual experience and estimation, but also on results from other research papers where other people have already tried to measure inattention (Zhang et al. (2020)) Furthermore, one must determine the computational question as different concepts of computing the probability of inattention exist. Our tool relies on statistical computations, as we divided the screen in parts and used time thresholds if someone looks at a certain zone for too long or not even on the screen. As mentioned above, also other concepts such as machine learning might also be well suited for detecting inattention. And last but not least, the design of a user-adaptive system plays an immense role, since the interaction with the user is the main purpose of the tool. Considering that we wanted someone to be attentive, we came to the conclusion that the interaction must be as minimalistic and plain as possible. We came up with different kinds of popup designs and after questioning some of our student colleagues, we chose the most minimalistic popup.

In the end, when developing a user adaptive system different kinds of difficulties and issues occur which must be handled in order to achieve a useful and properly working tool. The problems we came across are stated in the chapters above, however, there can be more things one has to consider while developing a similar tool. All in all and taking the limited resources into account, we believe the tool we have built serves as a good example of how an eye-based user-adaptive system can be developed and what further problems the developers have to solve throughout that process.

Bibliography

Broadbent, D. E. (1958). *Perception and communication*. Pergamon Press. https://doi.org/10.1037/10037-000

Csáji, B. C. et al. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, *24*(48), 7.

Dimoka, A., Davis, F. D., Gupta, A., Pavlou, P. A., Banker, R. D., Dennis, A. R., Ischebeck, A., Müller-Putz, G., Benbasat, I., Gefen, D., Kenning, P. H., Riedl, R., vom Brocke, J., & Weber, B. (2012). On the Use of Neurophysiological Tools in IS Research: Developing a Research Agenda for NeuroIS [Publisher: Management Information Systems Research Center, University of Minnesota]. *MIS Quarterly*, *36*(3), 679–702. https://doi.org/10.2307/41703475

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Evans, B. (2020). The Zoom Revolution: 10 Eye-Popping Stats from Tech's New Superstar. Retrieved August 15, 2022, from https://accelerationeconomy.com/cloud/the-zoom-revolution-10-eye-popping-stats-from-techs-new-superstar/

Fox, N. S., Brennan, J. S., & Chasen, S. T. (2008). Clinical estimation of fetal weight and the Hawthorne effect. *European Journal of Obstetrics, Gynecology, and Reproductive Biology*, *141*(2), 111–114. https://doi.org/10.1016/j.ejogrb.2008.07.023

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

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, *4*(2), 251–257.

Hutt, S., Krasich, K., Mills, K., Bosch, N., White, S., Brockmole, J. R., & D'Mello, S. K. (2018). Automated gaze-based mind wandering detection during computerized learning in classroom. *User Modeling and User-Adapted Interaction*, *29*, 821–867.

Kwon, K.-A., Shipley, R. J., Edirisinghe, M., Ezra, D. G., Rose, G., Best, S. M., & Cameron, R. E. (2013). High-speed camera characterization of voluntary eye blinking kinematics. *Journal of the Royal Society Interface*, *10*(85), 20130227. https://doi.org/10.1098/rsif.2013.0227

Lakens, D. (2022). Sample size justification. *Collabra: Psychology*, *8*(1), 33267.

Noordzij, M., Dekker, F. W., Zoccali, C., & Jager, K. J. (2011). Sample size calculations. *Nephron Clinical Practice*, *118*(4), c319–c323.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Shank, P. (2017). Attention And The 8-Second Attention Span. https://elearningindustry.com/8-second-attention-span-organizational-learning

Stiefelhagen, R., Yang, J., & Waibel, A. (2002). Modeling focus of attention for meeting indexing based on multiple cues. *IEEE transactions on neural networks*, *13*(4), 928–938. https://doi.org/10.1109/TNN.2002.1021893

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1202_4]. *Cognitive Science*, *12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Weber, W., & Ahn, J. (2021). COVID-19 Conferences: Resident Perceptions of Online Synchronous Learning Environments. *Western Journal of Emergency Medicine*, *22*(1), 115–118. https://doi.org/10.5811/westjem.2020.11.49125

Zhang, H., Miller, K. F., Sun, X., & Cortina, K. S. (2020). Wandering eyes: Eye movements during mind wandering in video lectures [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ac *Applied Cognitive Psychology*, *34*(2), 449–464. https://doi.org/10.1002/acp.3632

Affidavit

Ich versichere hiermit wahrheitsgemäß, die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des Karlsruher Instituts für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, 20.08.2022 J.Wessels, Marcel Namyslo, Paul Ferlitz, F. Wesch Anne Hoff

Joris Wessels, Marcel Namyslo, Paul Ferlitz, Anne Hoff, Fabian Weschenfelder

Prototype Video Publication Agreement

I hereby agree that the prototype video submitted by me may be published on the Internet.

Karlsruhe, 20.08.2022 J.Wessels, Marcel Namyslo, Paul Ferlitz, F. W Anne Hoff

Joris Wessels, Marcel Namyslo, Paul Ferlitz, Anne Hoff, Fabian Weschenfelder