Accelerating Knowledge Transfer by Sensing and Actuating Social-Cognitive States

Thesis approved by the Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.) to **Ko Watanabe**

Date of Defence:	November 22nd 2024
Dean:	Prof. Dr. Christoph Garth (University of Kaiserslautern-Landau)
Reviewers:	Prof. Dr. Shoya Ishimaru (Osaka Metropolitan University)
	Prof. Dr. Sebastian Vollmer (University of Kaiserslautern-Landau)
	Prof. Dr. Prof. h.c. Andreas Dengel (University of Kaiserslautern-Landau)

DE-386

Abstract

This thesis explores how smart sensors can quantify the process of the knowledge transfer. Knowledge transfer is the transmission of mastered knowledge from one individual to another through communication. This intricate process depends on three critical facets of communication: the appearance and demeanor of the participants, verbal articulation, and nonverbal cues. Several projects worked on will be to analyze the full potential of quantifying and enhancing communication.

Estimating individual domain knowledge is paramount to determining their ability to transfer knowledge. Web browsing analytics is a crucial in this endeavor, as it addresses the needs of our target audience to capture both new and existing knowledge. *TrackThinkTS*, an intuitive system, facilitates the seamless collection of web browsing logs by installing a Chrome extension. Our application using Random Forest algorithms to the collected data has resulted in an impressive average F1-score of 0.950 to estimate a domain knowledge. We extend this work to *TrackThink Camera*, allows collecting the webcam recordings synchronously while web browsing for appearance based eye-tracking. To enhance the visualization of respondent knowledge, we use force-directed diagrams, Sankey diagrams, and flowcharts, which are seamlessly integrated into the *TrackThink Dashboard*.

In the *DisCaaS* project, we aim to quantify micro-behaviors that happen during meetings using cameras as sensors. In collaboration with a research group in Japan, we meticulously collected a dataset of 295 videos, totaling 21.7 hours, with 40 participants from online and onsite meetings. Remarkably, we achieved F1 scores of 0.812, 0.949, and 0.973 for nodding, talking, and smiling detection, respectively. The *EnGauge* project has been instrumental in quantifying engagement levels, a critical dimension of internal and cognitive human behavior. We collect data from 30 participants and achieved a result of engagement detection of 0.895 in the F1 score with leave-one-participant-out cross-validation.

The concept of accelerating the knowledge transfer is done in several approaches. In the work *DiscussionJockey*, we apply dynamic background music and specific beats per minute change according to the meeting participants' utterance information to control amount of speech. We also investigate a system *Metacognition-EnGauge*. The system allows to give self-and-group engagement level feedbacks in gauge-interface realtime. Several challenges exist in the accelerating knowledge transfer, which will be discussed.

Applications have been shared with several labs, including the Immersive Quantified Learning Lab (iQL-Lab) at the German Research Center for Artificial Intelligence (DFKI), DFKI Lab Japan in Osaka Metropolitan University, Ubiquitous Computing System Laboratory (UBI-Lab) in Nara Institute of Science and Technology, and HumanoPhilic Systems laboratory in Kyushu University.

Acknowledgements

I am deeply grateful to Prof. Andreas Dengel and Prof. Shoya Ishimaru, my dissertation advisors. I sincerely thank them for giving me a great opportunity to work at the University of Kaiserslautern-Landau (RPTU) and the German Research Center for Artificial Intelligence (DFKI). They always give me insightful comments and suggestions.

I would like to give a warm thank you to Prof. Yutaka Arakawa and Prof. Koichi Kise. They have supervised me by sharing interesting research topics since I was a master student at Nara Institute of Science and Technology.

I would like to express my gratitude to Dr. Nicolas Großmann for his kind coordination at Immersive Quantified Learning Lab. I would like to thank Brigitte Selzer, Jayasankar Santhosh, Steffen Steinert, Prena Garg, Javier Carrasco Melo, Dr. David Dzsotjan, Dr. Jihed Makhlouf, Dr. Yuki Matsuda, and Dr. Yugo Nakamura for giving me valuable advices. They gave me insightful comments whenever I had problems.

I would like to acknowledge Aya Onishi, Dr. Andrew Vargo, Dr. Motoi Iwata, Dr. Benjamin Tag, Dr. Tilman Dingler, and Prof. Laurence Devillers for supports on the context of LeCycl project. I was inspired by their research ideas many times.

Most importantly, thanks to my students: Ankur Bhatt, Nabid Imteaj, Tanuja Sathyanarayana, Haruki Suzawa, Kanta Yamaoka, Sarah Gonzales, Kübra Kücük, Seiya Tanaka, Pooja Pol, Sahana Yadnakudige Subramanya, Haruka Sakagami, David Dembinsky, Riku Higashimura, Gitesh Gund, Anmol Ashri, Ryugo Morita, and Dai Shimizu. This thesis would not have been completed without their hard work and kind help.

Last but not least, my wife Yuika Watanabe. She always cheered me up when I was down and brought me a big smile to finish this thesis. Without her cheer and smile, I would not have completed this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the concept of learning-lifecycle, *Perceive*, *Master*, and *Transfer* are important factors [169]. *Perceive* is gaining knowledge from a textbook or lecture. *Master* is a process of perceiving knowledge usability by conducting exercises and experiments. *Transfer* transfers mastered knowledge to one another through a presentation or discussion. Figure 1.1 shows the visual image of the Learning Cyclotron (LeCycl). Analysis and acceleration of *Perceive* and *Master* have been done earlier [79]. However, analysis and acceleration of *Transfer* are yet to be discovered.

Understanding and accelerating knowledge transfer is essential in all kinds of fields. Allen *et al.* investigate that 11 million workplace meetings are held every day in the United States [7]. Employees spend an average of six hours in scheduled weekly meetings, with supervisors spending 23 hours [141]. Meetings are time-consuming for attendees and costly to the organization, ranging from 30 million to over 100 million US dollars annually [142]. Due to the importance of the field of study on knowledge transfer, the number of related publications is increasing every year [61]. Research on knowledge transfer is gaining attention and importance every year.

According to Shanon-Weaver Model [152], *Knowledge Transfer* is defined as communication between the sender (transmitter) and the receiver using encoding and decoding knowledge. Previous studies work on the implementation of a sensor environment for estimation of learning difficulty [162, 187], and estimation of wakefulness while taking a video lecture [90]. Our study aims to quantify and accelerate the success of encoding and decoding knowledge.

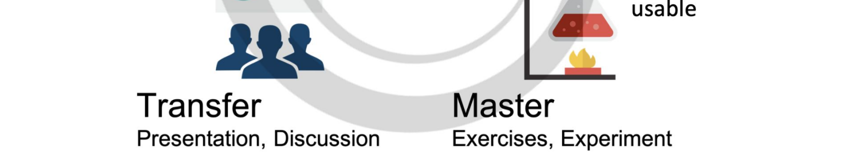


FIGURE 1.1: Overall concept of Learning Cyclotron (LeCycl) [169].

1.1 Basic Concepts

Knowledge transfer has been a cornerstone of human development throughout history. From the oral traditions of ancient civilizations to the sophisticated educational systems of the modern era, the ability to share and transmit knowledge has propelled humanity's growth and innovation. At its core, knowledge transfer represents more than the mere exchange of information; it is a process of understanding, adapting, and applying knowledge to create value across generations, disciplines, and societies.

In an era of rapid technological advancements, the importance of efficient and effective knowledge transfer has never been greater. In fields such as education, healthcare, engineering, and business, the ability to transfer knowledge determines individual success and drives organizational performance and societal progress. There is a need to promote seamless knowledge transfer across cultural, linguistic, and disciplinary boundaries.

However, the process of knowledge transfer also presents challenges. Communication, the basic medium of knowledge transfer, is strongly related to cognitive, social, and cultural factors. Misunderstandings and technical barriers often impede the flow of knowledge, leading to inefficient conversations. Furthermore, bridging the gap between experts and novices is a significant challenge as knowledge becomes more complex and specialized.

To address these issues, research in knowledge transfer must explore both the mechanisms of human communication and the tools that can facilitate this process. Emerging technologies such as artificial intelligence, sensor-based systems, and advanced analytics offer promising avenues to quantify and enhance knowledge transfer. These tools enable us to capture and analyze communication behaviors and provide actionable insights to optimize learning, collaboration, and innovation.

This paper explains the basic concepts of domain knowledge estimation, knowledge transfer visualization, and acceleration. The study seeks to unlock its full potential by leveraging state-of-the-art sensing technologies and intervention strategies. It contributes to a deeper understanding of how knowledge transfer can be systematically measured, improved, and applied to address the complex challenges of the modern world.

1.1.1 Factor of Knowledge Transfer

Recognizing the activities and social-cognitive states in communication is essential to understanding knowledge transfer. In communication, there are three key features: *Participants Appearance*, *Verbal Communication*, and *Nonverbal Communication* [49]. *Participant Appearance* represents features concerning the appearance of the participants in the communication. Schulte *et al.* found that in the communication scene, participant age and distribution have a significant positive relationship with forgiveness and the number of counteractive behaviors [149]. Due to the importance of participants' age for information transfer, Geng *et al.* has worked on age estimation from facial images using open source datasets such as the FG-NET Aging Database [100] and the MORPH Database [137]. *Verbal Communication* represents characteristics related to speech. Related research focuses on the analysis of conversational content [113], pronunciation [155, 188], speech volume [95], and voice pitch [112]. *Nonverbal Communication* refers to behavior-related characteristics that do not include speech. The related study focuses on the analysis of posture [16, 148], smile [30, 190], eye contact [182, 189], and head motion [128, 185]. Previous studies work on social activity behavior analysis in communication. However, some factors of knowledge transfer, such as micro-behavioral analysis and engagement detection, are still being studied.

1.1.2 Estimation of Knowledge Transfer

Estimating the state of knowledge transfer is presented by Srivastava [161]. The study works on the estimation of learning difficulty using contactless sensors. The study presents the use of the slider to collect a real-time understanding of the labeling. The contactless sensors are a Logitech webcam, a Tobii Pro X2-30 eye-tracker, and an Optris PI-400 thermal camera. After collecting data from 100 participants, the study found that the difficulty of the lecture depended on three factors: the pictorial representation of complex terms, the difference between the amount of verbal and visual information presented per slide, and the number of words spoken by the instructor per second [162]. Babaei *et al.* is then working on automatic tracking of attentional states using a webcam [10]. As a result, action units of the face, such as lips or eyelids, perform well in attention detection. Another work was done to discover the correlation between attention while watching videos and comprehension level [163]. Participants with higher prior knowledge made fewer eye movement transitions than those with lower prior knowledge. Prior knowledge estimation has been done by analyzing the web browsing logs [123]. Hence, the direction of web browsing analysis can perform well in knowledge transfer estimation. While the above studies deal with comprehension estimation, they focus on something other than the breakdown of comprehension into details of encoding and decoding in knowledge transfer.

1.1.3 Acceleration of Knowledge Transfer

Knowledge transfer is accelerated based on the state of knowledge transfer estimation results. Kuttal *et al.* work on human-human and human-agent comparison on pair programming [98]. The study discovered no significant difference in productivity, self-efficacy, code quality, and pair programming preferences when using an agent for pair programming. Test-driven programming can be done together with an agent. Knowledge transfer accelerates when the agent can provide code skeletons and hints to guide their partner toward a solution. Haliburton *et al.* use wristband devices to give thermal feedback to the presenter to feedback audience engagement [65]. The work could enhance socio-emotional connectedness in virtual and hybrid meetings. As a result, the presenter and audience feel connectivity only if they are in the same atmosphere, which enhances the presenter's confidence in transferring knowledge. Some studies work on the acceleration of knowledge transfer. However, there still needs to be more work on enhancing communication.

1.2 Research Questions

This thesis tackles three research questions. Understanding human activity and cognitive behavior for quantifying knowledge transfer is important to developing systems supporting humans. For that, the thesis came up with three research questions to answer.

How to discover the existing domain knowledge? Discovering humans' domain knowledge is significant in understanding people capable of delivering the knowledge. One question is whether the system can estimate humans' domain knowledge using sensing technology. One approach is capturing a person's web browsing behavior.

How to visualize the state of knowledge transfer? Knowledge transfer often happens in communication. Visualizing the important behaviors is important to understanding the success of the knowledge transfer. This thesis aims to visualize the significant behaviors for knowledge transfer.

How to accelerate the knowledge transfer? As a last question, the thesis aims to answer if an application can augment knowledge transfer. The objective is to implement intervention applications that improve communication and transfer skills.

1.3 Contributions

In summary, contributions of this thesis include:

- An overview of the state-of-the-art in activity recognition and intervention
- Methods to recognize activities in communication
- Methods to recognize engagement levels in meetings
- Methods to track eyes using webcam as a sensor
- Methods to recognize humans' affective states by using various sensors
- Development of a web browsing logger and the visualization system
- Development of meeting behavior visualization system

Applications in this thesis have been shared with several labs, including the Immersive Quantified Learning Lab (iQL-Lab) at the German Research Center for Artificial Intelligence (DFKI), DFKI Lab Japan in Osaka Metropolitan University, Ubiquitous Computing System Labo-ratory (UBI-Lab) in Nara Institute of Science and Technology, and HumanoPhilic Systems laboratory in Kyushu University.

1.4 Outline of Thesis Chapters

Chapter 2 presents an overview of activity recognition, engagement recognition, and interventions. Chapter 3 reports sensing applications for estimating and visualizing personal domain knowledge. Chapter 4 presents activity and cognitive recognition projects. Chapter 5 presents intervention applications for enhancing the quality of communications. Finally, Chapter 6 summarizes conclusions and future work.

Chapter 2

Background and Related Work

This thesis discovers an understanding of knowledge transfer. This chapter presents a literature survey about the background and related work. The chapter consists of three main sections. Section 2.1 looks at works on activity recognition. Section 2.2 looks at research on engagement recognition. Section 2.3 looks at intervention and application for knowledge transfer.

2.1 Activity Recognition

Figure 2.1 shows the overview of publications investigating key features' importance and detection algorithms in a meeting. The categories of important features can be separated into *Participants Appearance*, *Verbal Communication*, and *Nonverbal Communication*. Our target detection has yet to be proposed in previous research studies. We will highlight our contributions by showing the importance and progress of related works.

2.1.1 Participant Appearance

Participant Appearance represents features regarding the appearance of participants in meetings. Schulte *et al.* found, in meetings, participants' age and distribution possess a significant positive relationship with forgiveness and the amount of counteractive behaviors [149]. Forgiveness has been defined as "a reduction in negative feelings, and a recovery of positive feelings towards an offender after the offense has taken place" [87]. Counteractive statements had a negative impact on team meeting outcomes such as meeting satisfaction and team productivity [88]. Hence, the age of the meeting participant is an important factor to consider in the meetings. Geng *et al.* proposed method for age estimation based on facial aging patterns [59]. They used the open source database FG-NET Aging Database [54] and the MORPH Database [137] for the dataset. These datasets include people's ages and facial images. From the facial image, they extracted features by using the Appearance Model [48]. Then, using these features, they used SVM as the algorithm for estimating age. Concerning previous works, age detection or estimation from facial images or videos has already been explored [67, 75].

2.1.2 Verbal Communication

Verbal Communication represents features regarding *speaking* that occur during meetings. McDorman states that the context of what was discussed in the meetings is important [113]. Yu and Deng have proposed a method of automatic speech recognition (ASR) technique for

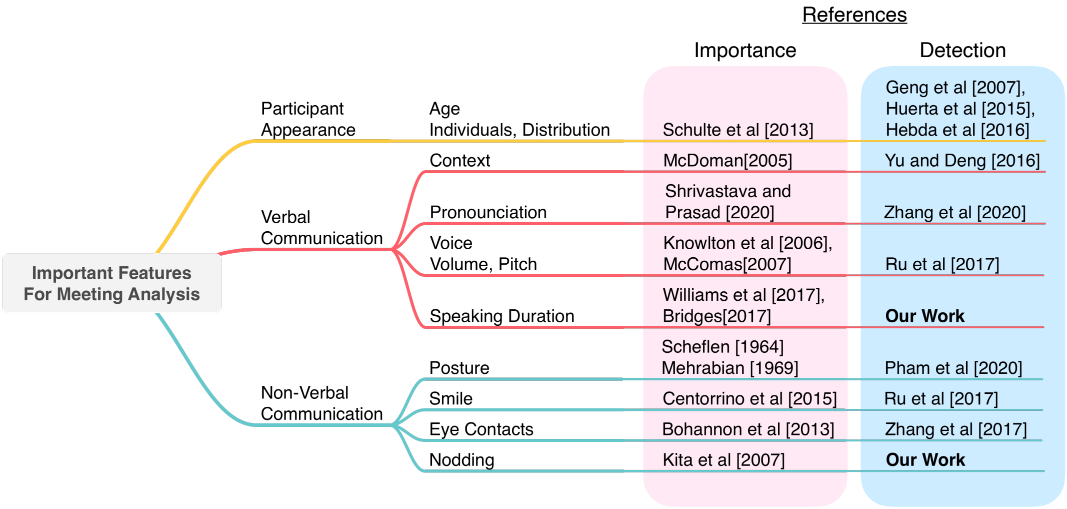


FIGURE 2.1: A tree map summarizing important activities for meeting analysis and the position of our work. The importance shows references representing the significance of each feature. The detection shows references from researchers who implemented the system to detect each feature.

speech to text [184]. Shrivastava and Prasad have stated that unclear pronunciation produce unclear understanding for listeners [155]. Precise pronunciation results in a high understanding of the context in meetings. In order to detect pronunciation errors, Zhang *et al.* have proposed a method of using deep learning techniques based on advanced automatic pronunciation error detection (APED) algorithms [188]. Knowlton and Larkin have found that voice volume and pitch can enhance anxiety or comfort listeners [95]. It has also been said that these factors also change participants' motivation to join future meetings in some cases [112]. Therefore, voice volume and pitch are important.

Zhao *et al.* have proposed the ROC Speak system, which allows ubiquitous access to communication skills training. They collect voice information from the microphone in order to detect the volume and pitch of the speaker [190]. However, this system has a limitation in that the user can only be collected as a single person. Sometimes, the voice cannot be collected during a meeting due to privacy issues. In order to tackle the idea of collecting verbal information concerning privacy issues, we focused on participants' *speaking duration*. Many research studies pointed out the issue of *manterrupting* [20, 179]. This paper suggests that men speak longer than women or cut off the topic when a woman is talking. In order to control the equality of turns, the collection of participants' speech duration is significant. We attempt to measure time duration by using only a camera as a sensor, which does not include microphones.

Janin *et al.* introduced the ICSI Meeting Corpus, which collects meeting speech logs with microphones. They have used both head-mounted and table-top microphones [85]. Carletta *et al.* introduced the AMI Meeting Corpus, using both cameras and microphones to collect meeting logs. They then created a transcript of meetings from the utterance data collected by microphones [27]. Riedhammer *et al.* then introduced automatic meeting summarization using the transcript produced from ASR [138].

The advantage of verbal data analysis is the ability to collect detailed data on what was spoken in the meeting. However, the disadvantage of this approach is the privacy risk. Verbal data can be a risk during collection for a company or in the education field. It includes private or confidential information that should not be recorded. Hence, verbal data analysis will always face a privacy risk.

2.1.3 Nonverbal Communication

Nonverbal Communication is related to features related to actions and does not include verbal information. Posture is one of the important nonverbal information in meetings [115, 148]. Pham *et al.* proposed 3D pose estimation from a single RGB camera. It can estimate body posture and activities by a camera [131]. Centorrino *et al.* have stated that a smile perceived as honest makes mutual trust and induces cooperation towards a person who sees the smile. With this in mind [30], Zhao *et al.* implemented the ROC Speak system in order to collect smiles automatically from the video [190]. Bohannon *et al.* have stated eye contact plays a role significant for human interactions [18]. Zhang *et al.* then proposed eye contact detection using a camera [189]. Ambient cameras and wearable glasses can be used. Kita and Ide have presented the significance of *nodding*. This study focuses on detecting *nodding* by a camera [94].

EKMAN and FRIESEN stated that nonverbal information contains emotions and the feature of interaction between multiple people [49]. Effective nonverbal communication facilitates meetings and allows speakers to make attractive statements. Moreover, it is easier for listeners to understand the statement. Morency *et al.* used a robot to collect human head movements by camera while asking questions. They aim to collect participant's head nods and head shakes. The recognition rate was 73% for head nods and 83% for head shakes [119]. Yu *et al.* used optical motion capture to collect nonverbal behaviors during a meeting. This approach allowed collecting *nodding* recognition of 76.4% and head shaking recognition of 80.0% [185]. Ohnishi *et al.* used the inertial measurement unit (IMU) sensor to collect participant's head movements [128]. They achieved the recognition of 97.5% in utterance, 52.4% in *nodding*, and 53.6% for looking around actions. The advantage of nonverbal data analysis is that it lowers the risk of privacy issues. It does not contain utterance information about each meeting. Moreover, it is an effective way of understanding emotions and interaction features between multiple people [49].

The disadvantages of previous research studies are the inconveniences and restrictions of the system settings. Connecting IMU sensors to the participants each time before a meeting is inconvenient. The preparation of a robot inflicts restrictions on where the meeting will be held. To create sustainable data collection or the sustainable use of the system, it must be simple, possess fewer devices, and have low restrictions on the use case environment.

2.2 Social Cognitive State Recognition

This section explains work on social cognitive state recognition. Section 2.2.1 shows work on engagement recognition. Then, Section 2.2.2 shows work on gender consideration in social-cognitive state recognition.

TABLE 2.1: Comparing EnGauge system with other engagement level prediction works.

Reference	Devices used	Participants	Labeling	Model	Results
Gao *et al.* [56]	Seat-Sensor	23 students	Self Report	N/A	N/A
Babaei *et al.* [10]	Webcam	15 researchers	Self Report	N/A	N/A
Mohamad *et al.* [117]	Webcam	20 students	Observer Annotation	VGG-B	Binary Classification 72.3% accuracy
Monkaresi *et al.* [118]	Webcam	23 students	Self Report	Naive Bayes	Binary Classification 75.8% accuracy
Hernandez *et al.* [68]	Wristband	51 children	Observer Annotation	SVM	Binary Classification 81.0% accuracy
Di Lascio *et al.* [45]	Wristband	24 students 9 teachers	Observer Annotation	SVM	Binary Classification 81.0% accuracy
DiSalvo *et al.* [47]	Wristband	11 students	Observer Annotation	SVM	Binary Classification 81.0% accuracy
Sharma *et al.* [153]	Webcam	15 students	Observer Annotation	CNN	Binary Classification 66.0% accuracy
Huynh *et al.* [76]	Smartphone Wristband Depth-Camera	64 students	Self Report	RF	Ternary Classification 77.0% accuracy
EnGauge	Webcam	24 students	Role-Acting	MobileNetV2	Ternary Classification 89.5% accuracy

2.2.1 Engagement Levels Recognition

In this section, we will elaborate on the types of devices utilized for engagement analysis and their accuracy level in predicting engagement. Table 2.1 shows the position of our work compared to the previous studies. The table shows the device each research study uses, the number of participants, how engagement levels are annotated, the machine learning model, and the prediction rate result.

Hernandez *et al.* use Empatica E4 wristband to collect children and adults engagement levels [68]. Using electrodermal activity (EDA) as an input and with SVM, it predicts the engagement level of around 81.0% accuracy. Di Lascio *et al.* use Empatica E4 wristband to collect participants' blood volume pulse, acceleration, peripheral skin temperature, and electrodermal activity while taking a lecture [45]. After the lecture, each participant was asked to answer a questionnaire to estimate their level of engagement. Using a Support Vector Machine (SVM) model, they achieved 81.0% accuracy in binary classifying participants' engagement levels. DiSalvo *et al.* worked on predicting student engagement level during lecture [47]. Participants were asked to wear an E4 wristband to collect electrodermal activity (EDA) signals. The subjects collected are eleven students between 18 - 30 years of age. The annotation for the engagement is done according to the specific behaviors defined. Using the SVM model, they got the binary engagement level classification result of 81.0%. These are the references to works using wearable sensors for detecting engagement levels. An interesting finding for these wristband usage approaches was that the accuracy of engagement prediction all scored around 81.0% when using the Empatica E4 wristband.

There are research approaches to using a webcam, not only wearable sensors. Mohamad *et al.* use a webcam to collect facial video and a convolutional neural network to binary classify users into engaged and disengaged [117]. The model contains two convolutional layers and two max-pooling layers with stride two and two fully connected layers, respectively. The last step of the CNN model includes a softmax layer, followed by a cross-entropy loss, which consists of two neurons indicating engaged and disengaged classes. With this architecture, they achieved an accuracy of 72.3%. Monkaresi *et al.* use a webcam for binary classify engagement levels. In the preprocessing, this work extracts facial action units [50] and heart rate [133] from the recorded video. Twenty-three undergraduate/postgraduate engineering students collect data from a public university in Australia. As a result, binary classification using Naive Bayes scored 75.8% accuracy. Huynh *et al.* worked on predicting engagement levels while playing smartphone game [76]. They collect smartphone touch actions, wristbands for photoplethysmography and electrodermal activity, and depth cameras for skeletal motion information. The ground truth of the engagement level is collected using a Game Engagement Questionnaire (GEQ). They use Random Forest (RF) as a best-fit model for the classification model. As a result, they achieved a ternary classify engagement level of 85.0%. Sharma *et al.* presented a system based on neural networks to detect the engagement level of the students captured by a typical built-in webcam present on a laptop computer, and their proposed system was designed to work in real time [153]. They combined eye, head, and facial movements to produce a concentration index with three classes of engagement: "very engaged," "nominally engaged," and "not engaged at all." Facial features were extracted using the Haar cascade algorithm from webcam images and then categorized by a convolution neural network (CNN). They tested the proposed system with 15 students in a typical e-learning experiment. They verified that the system could correctly distinguish between the three engagement classes with an accuracy of 66.0%. However, this study only focused on engagement level classification when watching e-learning video content. Hence, it did not cover the analysis of online meetings' behaviors.

Other than the studies above, some analyzed the correlation between engagement and collected sensor data. Babaei *et al.* classified low and high engagement levels using a webcam [10]. They annotated the level of engagement by using notifications to get labels from the participants. In this study, 15 participants' lab work data was collected. The camera device continuously records the participant's face while working. Using OpenFace to extract facial features, they found that the gaze angle and rotation in the vertical axis (pitch) significantly identify engagement levels. The study found that facial images can be good input data for predicting human engagement levels. Gao *et al.* worked on identifying the correlation between student engagement levels during lectures and the seating position [56]. They collected data from 23 students and discovered that students who sit close together are likelier to have similar learning engagement and tend to have high physiological synchrony.

2.2.2 Gender Consideration

In Human-Computer Interaction (HCI), the imperative of inclusive-gender design has been underscored [14]. The significance of gender manifests early in life, as evidenced by Cen *et al.*, who meticulously examined disparities among 110 toys designed for children aged seven or younger with a gender-oriented perspective [29]. The results unveiled a gender imbalance in coding kits, encompassing aspects such as color, physical form, and the activities involved. Cryan *et al.* has also noted distinctions in vocabulary or terminology between genders [40]. Notably, Song *et al.* discovered that male and female participants perceived high-pitched voices differently, adding a layer of intrigue to gender-related differences [159]. These gender nuances permeate various areas in HCI.

Although most of the work suggests that gender-inclusive designs are significant, Metaxa-Kakavouli *et al.* have found that gender-inclusive web user interface design can negatively impact women users [116]. Gender-neutral design was perceived positively by individuals of all genders. Also, Cryan *et al.* has mentioned the work on detecting software gender stereotypes using GenderMag [40]. The project aims to avoid or reduce gender stereotypes in software. Lastly, Hamidi *et al.* has mentioned that transgender individuals have overwhelmingly negative attitudes toward Automatic Gender Recognition (AGR) [66]. The system should consider privacy and potential harm from being incorrectly gendered or misgendered by technology. Concerning these researches, HCI needs to be careful in considering gender as a feature that does not always act reasonably, and hence, whether it performs well needs to be investigated.

Brody dives deep into how emotional development varies between genders [22]. It sheds light on the unique paths boys and girls take in emotional processing, emphasizing the decisive role of socialization and sociocultural factors. This understanding is critical, especially as we delve into neural network-based predictions of affective states. The work explored methodological challenges, and the potential of deep learning in this gender-focused emotional research is a cornerstone for our work in developing predictive models. Building on the theme of gender and emotional expression, the study by Andonova and Taylor delves into the intriguing world of non-verbal communication across cultures. Their research reveals how head movements, commonly used to express agreement or disagreement, can vary significantly between cultures, such as in the United States and Bulgaria [8]. This highlights the profound influence of cultural norms on emotional expression and interpretation, offering valuable insights into how gender may intersect with cultural practices to shape emotional communication.

Complementing our exploration of gender differences in emotional expression, the Affectiva-MIT Facial Expression Dataset (AM-FED) by McDuff *et al.* presents a groundbreaking resource for studying natural and spontaneous facial expressions [114]. This dataset, which includes facial videos collected in the wild via webcams, offers a diverse and ecologically valid sample of responses to online media. This dataset's detailed labeling of facial action units, head movements, and self-reported emotional experiences provides an invaluable tool for examining the subtleties of emotional expression across different genders in real-world settings.

2.3 Intervention and Applications

This section explains work on intervention and application for actuating social-cognitive states. Section 2.3.1 shows the intervention of the meeting-encouraging system. Then, Section 2.3.2 shows works on social-cognitive state augmentation.

2.3.1 Meeting engagement encouraging system

Encouraging participants' engagement in meetings has several works [111, 183]. Matsuyama *et al.* proposed a method using a facilitation robot to enhance engagement levels in offline meetings [110]. They set a condition for encouraging four participants, including the robot, in each set of experiments. The robot uses the mounted camera to discover the engagement status of the other three participants or how well they are harmonized with other participants. The study showed that instead of three participants having conversations, adding a robot as a fourth participant in the meeting was influential in encouraging all participants to join in the conversations. Adriel Aseniero *et al.* proposed the MeetCues system [4]. This system visualizes the emotions of attendees during online meetings. Participants click the like and clarify buttons to choose their status. The system then shows each participant's emotions, such as happiness, neutrality, and thinking. An emoji presents each emotion so other users can recognize the participant's status from the interface. Gashi *et al.* target on measuring physiological synchrony, which is the synchronization of the physiological states of multiple people [58]. The research shows that when multiple people interact in synchrony, the engagement level toward the content presented is high. Hence, they aim to implement a system for giving feedback when synchrony happens in the lecture or conversation. It will support the teacher or presenter understand when and which slide interests the students or audience.

2.3.2 Social cognitive state augmentation system

Holstein *et al.* has proposed a method for the teachers to check the students' cognitive states as a dashboard [70]. In the dashboard, each student has an indicator display floating above the head. The work proposed visualizing multiple people's cognitive states combined as a dashboard. *Jumple* project aims to augment the virtual physical education experience [154]. The work supports visualized interactivity with remote participants. Niwa *et al.* state the importance of AI agents to augment human cognitive abilities [127]. Their approach is to implement a similar-looking AI avatar as a participant to investigate the trustworthiness level of agent statements. Lastly, Kytö *et al.* has proposed a method for visualizing public and personal information using digital profiles [99]. The work states that giving meta-information about a person communicating will support smooth face-to-face communication. Concerning previous research, augmentation of the cognitive state positively supports change in behaviors and understanding of humans.

Chapter 3

Domain Knowledge Estimation

This chapter explains the process of estimating human domain knowledge. Discovering domain knowledge can support finding a potential knowledge sender. Also, it may be a tool to verify the domain knowledge of a receiver after receiving new knowledge.

Section 3.1 explains *TrackThinkTS* [107], the web browsing logger application. The application works by installing a web browser as an extension. This tool can collect user web browsing actions such as *TabCreate*, *TabActivate*, *TabUpdate*, *TabRemove*, *ClipboardCopy*, and *WindowScroll*. The experiment will collect experts and novices while conducting programming tasks using a tool. The analysis is achieved by classifying two groups. Also, force-directed diagrams and Sankey diagrams will be presented for further visualization.

Section 3.2 explains *TrackThink Camera* [176], which add-on the camera into the *TrackThinkTS*. The application is extended to understand participants' affective states while seeking information while browsing. This application strengthens all-in-one technology for tracking web search behavior and synchronizing webcam recordings.

Section 3.3 [17] explains the approach for webcam-based eye-tracking. The research aims to detect gaze coordinates from the webcam images. The approach supports extending *TrackThink Camera* to collect gaze data while web browsing.

Section 3.4 explains *TrackThink Dashboard*, the GUI application for visualizing logging data of *TrackThinkTS*. The application supports the visualization of the participant's workflow during programming. The tool supports understanding how individuals seek information and apply outputs.

Overall, this chapter contributes to showing new implementations for domain knowledge estimation. Targeting specifically on programmers, achieving the classification of experts and novices. A deeper analysis of the workflow has also been evaluated.

3.1 TrackThinkTS: Web Browser Extension to Track User Search Behavior

In a digital world where an immense load of data is created every second, knowing where and how to search for the correct information is essential. Terms such as “Information Overload” are frequently used to express the phenomena [69]. Moreover, individual web search behavior and abilities became topics of meticulous research [12, 43, 74, 91].

Additionally, the problem of finding the correct information is amplified when searching for programming errors, examples, and code snippets [144]. Traditional general-purpose search engines are not optimized for programming code search [180]. Instead, they are developed to handle natural text requests. Thus, sometimes, they need to recognize the context and semantics of the code search [134]. Several tools were created to address the code search problem like the Google Code Search [1], Source Graph [2], Searchcode [3] and many others. However, some tools were either discontinued or obsolete [134].

Undoubtedly, web searching is integral to the software programmer’s activities. The purpose of the web search can vary a lot depending on the context. However, many studies found that web searching is one of the most frequent activities of software developers [71, 144, 180]. However, most of the studies about code search were conducted in a professional environment. The participants in these studies had different levels of expertise in programming. Less focus was given to investigating the web searching behavior of programming students.

Students use search engines for various purposes while learning programming, whether self-learning or in academic settings. In addition, the recent events related to the COVID-19 pandemic made the situation even worse. Educational institutions had to shift to fully online learning. This reduced considerably the opportunities for students to communicate and collaborate. Hence, they lost the chance to learn from each other. Moreover, interactions with professors and teachers are needed to gain face-to-face communication. In the programming case, teachers used to monitor and help students more effectively. Industry and academia are trying to mitigate the negative effects of the lockdown. Nevertheless, students in general, and programming students in particular, rely more than ever on online resources. It is crucial to help them acquire the skills that help them achieve fulfilling web searches. Nevertheless, a few research works focused on this sub-population of programming web searches. Therefore, this study introduces a tool for examining students’ web search behavior while they solve programming exercises.

In the current manuscript, we present a browser extension called TrackThinkTS. This extension logs the browser usage while searching and surfing on the web. It was developed with a privacy-first mindset. Users can view, edit, and delete entries before exporting the data to CSV format. Along with their full consent, participants in the experiments using this extension have an aggregated view of the stored logs for easier management. This extension will

[1] https://developers.google.com/code-search

[2] https://sourcegraph.com/

[3] https://searchcode.com/

serve as the building block for subsequent experiments and analyses that aim to investigate the following research questions:

- Is there any difference in web search skills between successful students and the others?
- To what extent this difference in web search skills influences the learning?
- What are patterns of effective web search skills manifested through thought processes?
- What are the best ways of sharing thought processes in course supplements?

3.1.1 Architecture

Many tools track the users' web searches. However, most of them only capture the pages visited using the URL and provide some analytics and visualization based on that [11, 46, 120, 181]. However, this approach needs to grasp the whole web search behavior of the user. In addition, most modern browsers provide a complete API for sophisticated access to the internal browser state and allow advanced manipulations. The tool presented in this study is the continuation of a previously discontinued proof-of-concept by the same name of TrackThink [123]. It was unfinished and had a few problems. The main objective was to gather as much information as possible about the users' actions within the browser when they engage in web search to capture their "thought process". Compared to the previously mentioned tools, it did not rely solely on the history of the web pages and URLs. However, it was not published in the Chrome extension store and was used only by the developers. The UX needed many improvements. Moreover, it was not optimally adjusted for user privacy or user convenience. Therefore, in the present update, we aim to transform the proof-of-concept into a fully working product and improve several critical aspects such as:

- Improve the user privacy by giving full control to the user on the registered logs.
- Improve the UX and workflow for both the participants and the experiment organizers.
- Publish it in the Chrome store for easier distribution and installation to students.

Ultimately, TrackThinkTS inherits the name and some concepts of the original unfinished work, but it is fundamentally different in many aspects, including the generation of logs, storage, and the whole workflow and experiments.

Logs Generation

Previous work interested in user behavior when searching the web used limited information. They gathered the history of the web search, represented mainly by URLs. However, in our case, we would like to gather as much data as possible by exploiting the browser capabilities and API. TrackThinkTS captures several events within the browser along with the visited page URLs. Remarkably, the interest in gathering tabs management is driven by the nature of the use case. The users have to switch between tabs frequently. They use some tabs to search

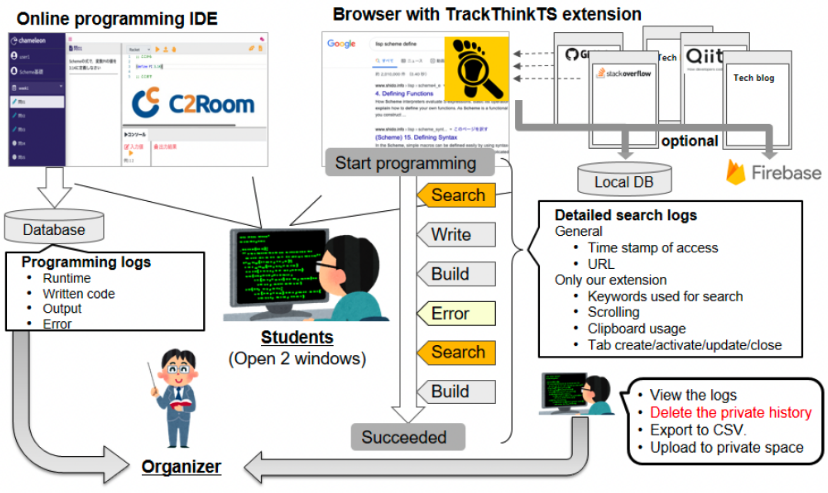


FIGURE 3.1: TrackThinkTS workflow overview.

the web, then return to the online IDE, where they continue working on the programming exercises. Therefore, tracking systems based on URLs could be more effective.

Tabs Management: There are four particular events related to tabs that TrackThinkTS monitors: Tab creation, Tab activation, Tab update, and tab delete. There are many advantages to using the browser extension Tab API. Tab creation does not carry much information, but it can be useful to detect which is the user's default new tab page and to detect advanced users. Tab activation is important to collect data about page switches. Tab update is triggered when a change happens to the page loaded in a tab. It can be visiting a new page in the same tab, a refresh, or loading some new content. This is particularly useful to capture events that do not require a full page reload (e.g., Javascript events). These events would be challenging to detect if we rely only on the history and URLs. Finally, Tab delete is triggered when the user closes a Tab.

Window Operations: The extension also monitors users' specific actions, namely Scrolling and Clipboard usage. When the user scrolls on a page, we gather the corresponding coordinates and the page's visible content. Additionally, we collect information about the viewport of the page. This way, we can quickly recover which part of the information displayed on the web page was the most helpful. Additionally, if the user copies pieces of text, we detect it and save the copied text. Every captured event is also appended to some additional information, such as the timestamp and user identifications. We assign each user a randomly generated user ID and ask the participants to input their names.

Workflow

The development of the TrackThinkTS extension was achieved using the browser extension API. TrackThinkTS officially supports the browsers like Google Chrome, Brave, and Vivaldi. Any other chromium-based browser could use the extension, but we did not perform thorough testing. Figure 3.1 shows the workflow of TrackThinkTS. There are two ways of storing the log data. The first method is to use a cloud-based database such as Firebase. This storage option is deactivated by default and only available when the experiment is conducted in a controlled environment (more details in Section 5). The second storage option is to save the log data in local browser storage. Later, the logs are aggregated, and the user can manage the logs within an integrated dashboard of the extension. After checking the logs, the user can export them into a CSV file and submit them to a submission location prepared by the experiment organizers. A proper manual was prepared for the users [4]. The typical workflow will be as follows. Students will be solving programming problems and can search the web. They write their code in an IDE; when they are stuck or encounter an error, they turn to the Internet for help. Meanwhile, the extension logs most of their actions within the browser. Conversely, the researchers and experiment organizers can access the logs from two locations. If the experiment were conducted in a controlled environment with the cloud database upload activated, they could access the dataset directly. Otherwise, the researchers must prepare the appropriate structure to store the students' submissions before using them.

Logs Control

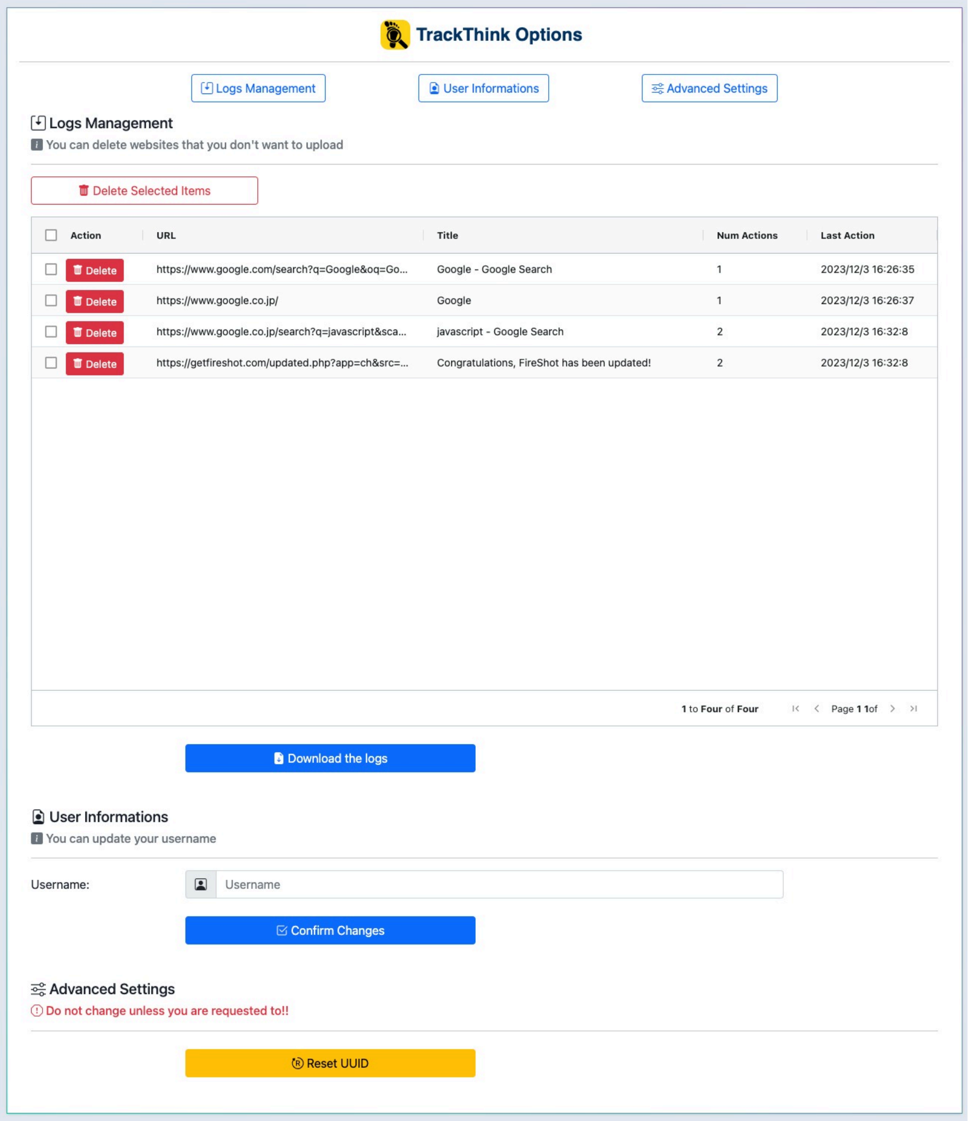
The TrackThinkTS extension logs almost all the users' actions in the browser. Therefore, there is a natural concern about the users' privacy. To address this issue, we give the users complete control over what they want to submit. There are two levels of control. The first level of control happens within the extension itself, and the second level of control occurs when the user exports the logs into CSV and can manage the dataset as it is. In fact, on the TrackThinkTS configuration page, we display the list of all actions aggregated by URL. That means each row represents a unique URL. The row includes the number of actions executed on that page.

Figure 3.2 exposes the TrackThinkTS application view. All the aggregated data logs are displayed in an advanced data table. Based on this aggregation, the users can delete all the logs saved for a particular URL. However, the logs need to be shorter. So, the users can reorganize and order the entries or apply advanced filters to find specific URLs and remove the corresponding logs. URLs are unique but might share the same domain (e.g., google.com), so filtering is useful.

The options page also allows the participants to input their names, but more importantly, it allows the experiment organizers to apply advanced settings, which activate the cloud upload of logs and the rest of the randomly generated user ID. A password protects these settings, so users cannot tamper with them.

The TrackThinkTS browser extension aims to be a foundation for several studies on web search behavior and course supplements. We will mainly focus on web search behavior in the case of programming learning and assignments. Two experiments have been recently performed using TrackThinkTS. The first experiment was conducted in a controlled environment where participants had to use a dedicated computer in a reserved room. All the necessary tools, including TrackThinkTS, were installed on the computer. In this controlled environment, the logs were gathered using a cloud-based database, and the participants had

[4] `https://ubi-naist.github.io/TrackThink/en/usage`



Logs Management

You can delete websites that you don't want to upload

□	Action	URL	Title	Num Actions	Last Action
□	Delete	https://www.google.com/search?q=Google&oq=Go...	Google - Google Search	1	2023/12/3 16:26:35
□	Delete	https://www.google.co.jp/	Google	1	2023/12/3 16:26:37
□	Delete	https://www.google.co.jp/search?q=javascript&sca...	javascript - Google Search	2	2023/12/3 16:32:8
□	Delete	https://getfireshot.com/updated.php?app=ch&src=...	Congratulations, FireShot has been updated!	2	2023/12/3 16:32:8

1 to Four of Four Page 1 1of

User Informations

You can update your username

Username: Username

Advanced Settings

Do not change unless you are requested to!!

FIGURE 3.2: TrackThinkTS application view.

to export the logs manually. This was the setup before the TrackThinkTS extension was accepted for publishing in the Chrome extension store.

The second experiment was conducted on a larger scale after the extension was available in the store and participants could use their computers. Therefore, the cloud-based logging using Firebase was disabled. Both experiments consisted of a series of Functional Programming exercises. The participants used an online educational software called C2Room [5] that included an online IDE for programming. Within the C2Room software, they can access the exercises and start solving them using the built-in IDE. Each one of the experiments consisted of ten exercises using the Scheme programming language with increasing difficulty. Participants were told they were free to search for online resources when needed.

[5] `https://c2room.jp/`

TABLE 3.1: TrackThinkTS logs detail

Action Related	Details
UUID	An unique user ID generated for each user.
User Action	*TabCreate*, *TabActivate*, *TabUpdate*, *TabRemove*, *ClipboardCopy*, and *WindowScroll*.
Datetime	Is the timestamp of the action performed.
Tab Related	
Title	It is the title of the page where the action happened.
URL	It is the URL of the page where the action happened.
BodyText	It is the whole content of the page where the action happened.
Contextual	
Scroll	It is data not filled here unless the *User Action* is a *WindowScroll* event. The data contain viewport, the speed and rate, and the document width and height.
Clipboard	It is data not filled here unless the User action is a *ClipboardCopy* event. The clipboard data basically contain the copied text.

TABLE 3.2: C2Room logs detail

Action Related	Details
UID	The user ID set up before the start of the experiment.
ClassID	The ID of the class to which the user enrolled.
TaskID	The ID of the task that the student is solving.
Time	The timestamp of the action performed.
OP	The operation type that the user has made. *ConID*, *MoveTask*, *StartCompiling*, *EndCompiling*, *SubmitCode*.
Data related to compilation start	
Code	It contains the programming code to be compiled.
Lang	The environment that will handle the compilation of the code.
stdin	The input to the compilation phase.
Data related to compilation end	
Response	It basically says if the compilation succeeded or failed. Therefore, its values are: success or error.
Startime	The timestamp when the compilation started.
stdout	The output of the code after compilation and execution.
stderr	The error message if the compilation fails.

Dataset composition

As explained earlier, the TrackThinkTS browser extension logs different types of events related to browser usage. Moreover, the students used the C2Room educational software containing an online IDE to solve the programming exercises. We could also recover the students' usage logs from the C2Room software. Furthermore, we could gather additional information about the students participating in the experiments. Accordingly, the overall dataset is rich and diverse. Each recording in the dataset contains various information depending on the source. *TrackThinkTS* information represents an actions defined in Table 3.1. *C2Room* information represents an actions defined in Table 3.2. Student information represents actions defined in Table 3.3.

TABLE 3.3: Students information detail

Survey	Details
Basic Information	Name, Email address, Age, Gender, and the assigned userID.
Programming Experience	Years of programming.
	Familiar Programming languages.
	Average number of days per week doing programming.
	Several questions about how they deal with programming errors.
Feedbacks	About the experiment and the exercises.
Score	Students' exam scores in the programming course.

Objectives

Using TrackThinkTS, we want to track the students' search behavior and thought process when they have to solve programming problems. Firstly, we want to investigate the difference in successful web searches between the high-performing students and the others. Based on the diverse studies related to web search behavior [134, 178, 180], cognitive abilities and skills related to web search expertise influence the success of the web search. Such meta-knowledge and search expertise can be shared and transferred [121]. Therefore, we aim to find the patterns of search expertise expressed by successful students and transfer them to less successful students using course supplements to help them acquire the tools for independent self-improvement. Using the exam score and the dataset from TrackThinkTS, we can compare the search behavior of successful students with that of others. Moreover, in our case, measuring web search expertise is accomplished by finding successful web searches in the context of programming errors. Accordingly, we also acquire the online resources used by the students to solve their programming errors. Thanks to the fine-grained data from TrackThinkTS, we can store the exact information or code snippet that helped the students solve their errors. If we synchronize this type of data with the stack trace we recover from the C2Room IDE, we can provide timely help to future programming students.

Data Aggregation and Synchronization

So far, the research experiments have been conducted using an online IDE. It is possible to recover the students' build status and execution states and synchronize them with the web search logs. This allows us to capture the successful build and accurately recover which text or page was the most helpful to the students. In addition, refined data aggregation using the scroll behavior and timestamp can lead to advanced metrics. These metrics were used to study users' browsing strategies [106]. We are also interested in using them in students' web search behavior. Bounce is a user's interaction that is considered a bounce when the engagement time is relatively short, and they leave (either switch tab or close it) quickly. Shallow engagement refers to the user reading less than half of the page's content. Deep engagement implies that the user reads over half of the page's substance. Complete engagement is the strongest when the user reads most of the content and decides to save it as a bookmark, keep the tab open, or save it elsewhere.

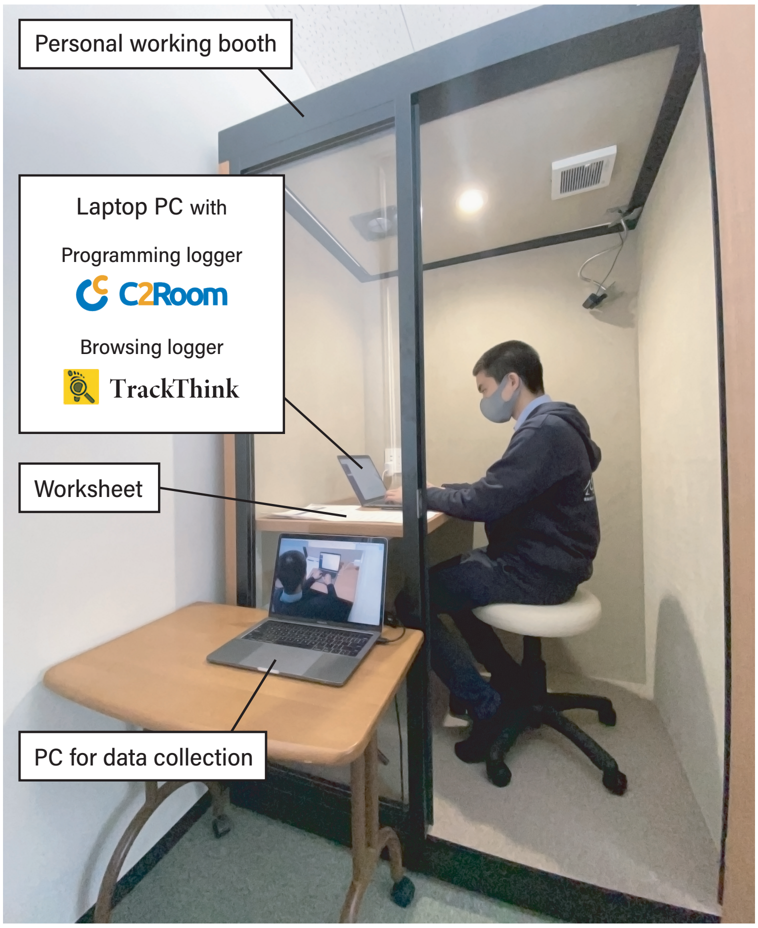


FIGURE 3.3: Experiment setting. Collect programming and browsing log while solving questions.

3.1.2 Experiment 1: Domain Knowledge Estimation

Here, we show findings of domain knowledge estimation [174]. The experimental settings are shown in Figure 3.3. C2Room and TrackThinkTS were installed on the participants' laptops before the experiment. Under this condition, the experiment was conducted using the procedure shown in Table 3.4.

Task Setting

In this experiment, we selected *Scheme (Racket)*, which is one of the dialects of LISP languages, as the programming language of the task [3]. These reasons for selection are (1) It is employed in programming lectures of several universities because of its simple language specification, and (2) It is customized for the usage of lectures. Hence, its language specification is only usually known by students for lecture attendees. Table 3.5 shows the questions they are asked to answer. We have prepared ten questions, which are sorted by their difficulty. These questions are selected on our own. Easy questions require less code and an understanding of the domain knowledge. These questions are simple to answer for participants with domain knowledge. However, participants who needed domain knowledge were required to search for the basic rules of the scheme. Therefore, we set these questions to classify users with and without domain knowledge.

TABLE 3.4: Experiment procedure for domain knowledge estimation

Process	Details
1	Explanation about the purpose of the study, experimental settings, and tools is provided to the participant. Only the person who is agreed with the conditions can participate this experiment.
2	The participant enters the personal working booth, and both programming and browsing logger will be started to record data.
3	The participant solves given scheme questions on the programming editor of C2Room. We did not restrict the order for answering questions, although supposing the participant will start from the easy part. The result will be recorded by C2Room at every compile execution.
4	When the participant faces unknown syntax and/or compile errors, they will search in the web. This behavior will be recorded by TrackThinkTS.
5	If compile result seems correct, the participant submit the answer and proceed to next question. The participant can correct their answer anytime by returning to previous answered questions.
6	Until finishing to solve all questions or elapsing one hour, the participant repeat procedure 3–5.

TABLE 3.5: Scheme Questions

ID	Question
1	Define variable *PI* as 3.14.
2	Write a scheme to show PI $*5^2$.
3	Write a scheme to show $(-b+\sqrt{b^2-3ac})/3a$.
4	Define function *areaDisk* to calculate circle area from radius *r*.
5	Define function *areaRing* to calculate circle area from outer and inner diameter *D, d*.
6	Define function *d2y* that convert US currency dollar *d* to Japanese currency yen. Note that 1 US dollar is 108.43 yen.
7	Define function *e2d* that convert European currency *e* to United states dollar. Note that 1 euro is 1.1069 US dollar.
8	Define function *p2e* that convert British currency pond *p* to European currency. Note that 1 pond is 1.1632 euro.
9	Define function *p2y* that convert British currency pond *p* to Japanese currency. Use *d2y*, *e2d*, and *p2e* in the previous questions.
10	Define function *c2f* that convert Celsius *C* to Fahrenheit. Note that $f = 1.8c + 32$.

Dataset and participants

We conducted experiments with two groups: novice students (Dataset A) and lecture attendees (Dataset B). We experimented with two types of groups to discover and compare how the difference will occur in each group. Here, we explain the details of each dataset. Our experiments do not include EU citizens; hence, the General Data Protection Regulation (GDPR) is not applied to our recordings. We define domain knowledge as the participants

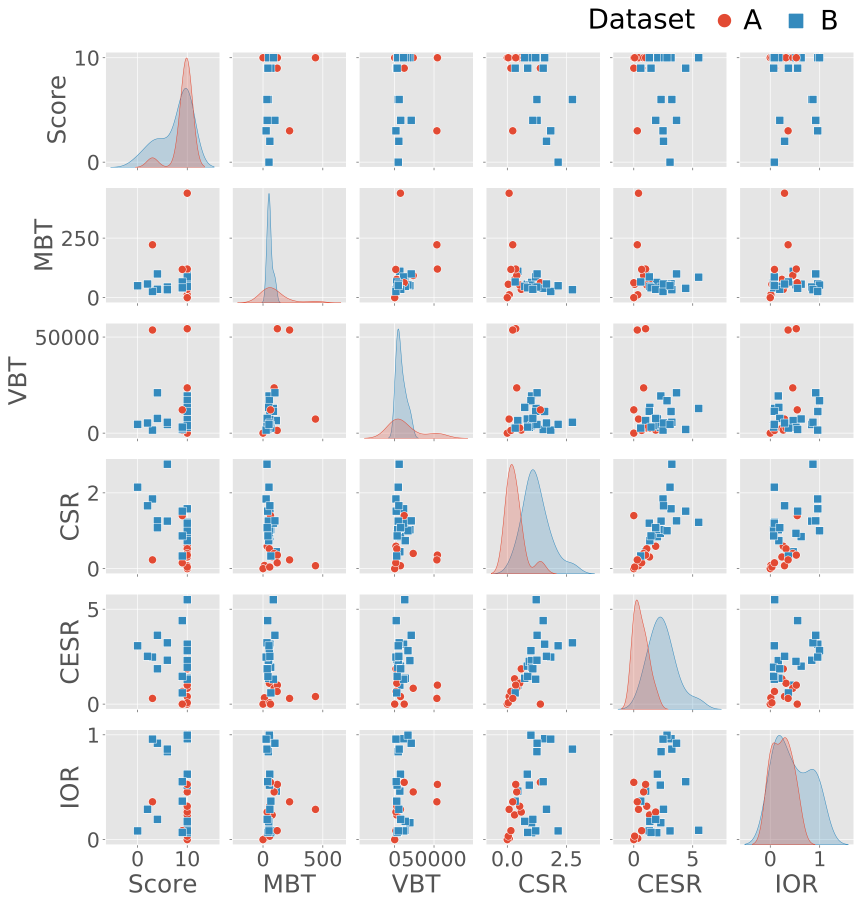


FIGURE 3.4: The pair plot shows the semantic mapping of datasets A and B. Dataset A is a group with domain knowledge, and Dataset B is a group without domain knowledge. This plot notes that CSR and CESR have unique peaks in the histograms of each dataset.

who took classes related to the Scheme in the university lecture. These reasons for selection are (1) It is employed in programming lectures of several universities because of its simple language specification, and (2) It is customized for the usage of lectures. Hence, its language specification is only usually known by students for lecture attendees.

Dataset A – group of lecture attendees (With domain knowledge) Data were collected from 13 unique participants (12 males and one female). All participants took lectures on the Scheme. Hence, they have some domain knowledge of the programming language's grammar.

Dataset B – group of novice students (Without domain knowledge) Data were collected from 20 unique participants (20 males). None of the participants took lectures on the Scheme. Hence, they do not have domain knowledge and are required to search for the programming language's grammar.

Evaluation metrics

Here, we present evaluation metrics used for the evaluation.

Domain Knowledge (DK) – Estimating whether a user has domain knowledge is one of our main objectives. This ground truth value can be obtained from not two software but an experimental setup, such as recruiting two participant groups (with and without DK) or conducting a pre-test.

Score – The Score of the programming problems is calculated from the log of C2Room and represents task-specific knowledge or problem-solving levels. In addition to DK, we also compare this value with other metrics.

Mean and Variance of Browsing Time (MBT, VBT) – We define Browsing Time (BT) as the time difference from *tab activate* until the last *window scroll* on a web page collected by TrackThinkTS. We store this value for each page and calculate their means and variances as metrics.

Compile Search Ratio (CSR) – How often a user searches a website to solve one problem can be calculated from the combination of C2Room and TrackThinkTS. We define this metric as follows.

$$CSR = \frac{\text{Number of web page access}}{\text{Number of compiles}}$$

Compile Error Search Ratio (CESR) – CESR is an extension of CSR that focuses on only the number of incorrect complies instead of all.

$$CESR = \frac{\text{Number of web page access}}{\text{Number of compile errors}}$$

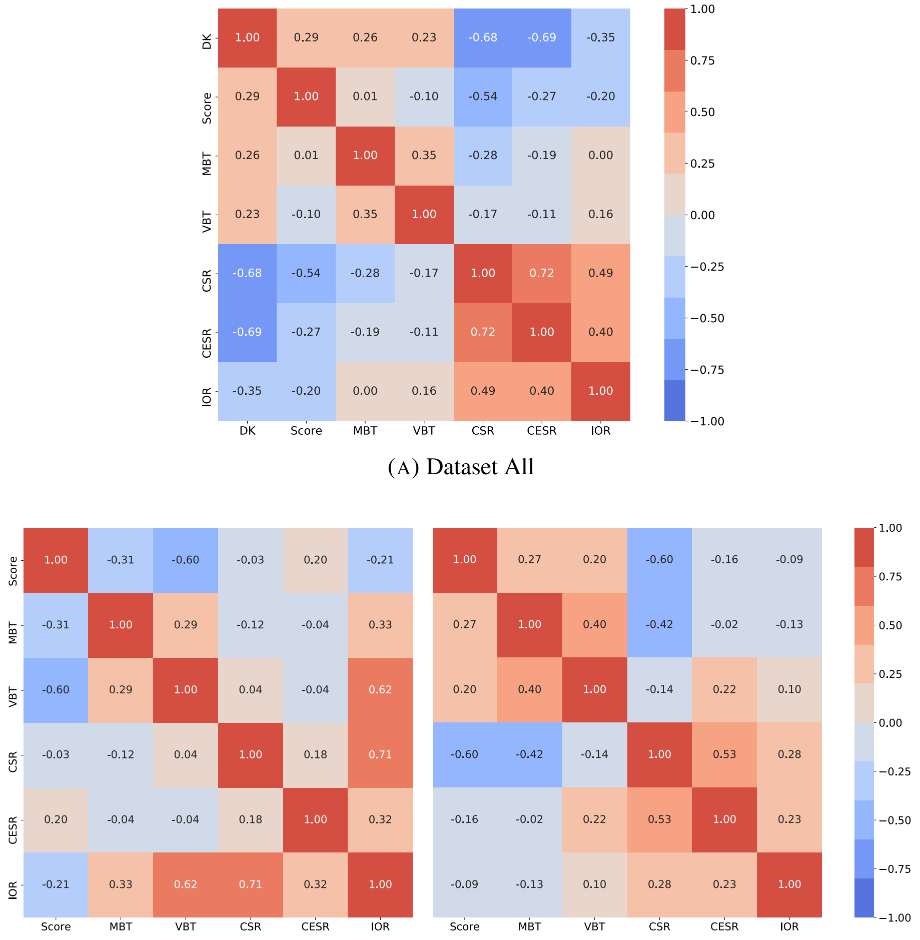
Input Output Ratio (IOR) – We are interested in how much time a user spends on input and output for a given task. In the case of this browsing programming, we define the indicators as follows.

$$IOR = \frac{\text{Time duration of search}}{\text{Time duration of search} + \text{Time duration of coding}}$$

3.1.3 Results and Discussions

This section explains the result of the experiment. Figure 3.4 shows the pair plot group by each dataset A and B. Dataset A is lecture attendees with domain knowledge, and Dataset B is the novice students. Section 3.1.2 shows the calculation methods for all metrics. The result confirms that CSR and CESR have different peaks in the histogram. Looking at the histogram of MBT and VBT, there is no significant peak for dataset A. In dataset B, with IOR, the histogram's peak values indicate two characteristic groups. Lastly, looking at the pair plot of CSR and CESR, datasets A and B have different groups. Dataset A is grouped in low CSR and CESR, and dataset B is in high CSR and CESR.

Next, the correlation of each metric in detail. Figure 3.5a shows the correlation heatmap including all participants. The significant correlation with DK is CESR with -0.69 and CSR with -0.68. Also, the score and the CSR have a negative correlation of -0.54. CESR and CSR were significant indicators of domain knowledge and score. Figure 3.5b and Figure 3.5c is the heatmap for each dataset A and B. For dataset A, score and VBT have a negative



(A) Dataset All

(B) Dataset A

(C) Dataset B

FIGURE 3.5: Correlation heatmap of dataset.

correlation of −0.60. For dataset B, score and CSR have a negative correlation of −0.60. The key metric for obtaining a high score was different for each dataset.

Lastly, the result of predicting the existence of the domain knowledge. Table 3.6 shows the prediction rate of the domain knowledge. The highest mean accuracy was 0.95, as recorded by Random Forest, with a standard deviation of 0.15. The parameter setting of the model is hyper parameters *the number of trees: 100*, *Criterion: Gini impurity*, and *the number of max features: 7 (square root of the number of features)* are used for classification. The lowest mean accuracy was the Support Vector Machine. Table 3.7 shows the result of feature importance for the model using Random Forest. The most important input was CESR, and the least important was IOR.

RQ1: Do participants with and without domain knowledge each have similarities in web browsing and programming activities?

Figure 3.4 shows the pair plot with a semantic mapping of dataset A (with domain knowledge) and B (without domain knowledge). According to the histogram, the pattern of CSR and CESR in the red square is similar for each dataset A and B. Also, by combining CSR and CESR as features, it is possible to classify people with and without domain knowledge.

TABLE 3.6: 10-fold cross validation result of binary prediction of lecture attendees and novice student.

Machine Learning Model	Mean Accuracy	Standard Deviation
Logistic Regression	0.90	0.20
Linear Discriminant Analysis	0.70	0.40
K-nearest Neighbors Vote	0.70	0.33
Decision Tree Classifier	0.85	0.32
Random Forest Classifier	**0.95**	**0.15**
Gaussian Naive Bayes	0.85	0.23
Support Vector Machine	0.60	0.49

TABLE 3.7: Feature importance of Random Forest Classifier.

Rank	1	2	3	4	5
Feature	CESR	CSR	MBT	VBT	IOR
Weight	0.50	0.24	0.13	0.07	0.05

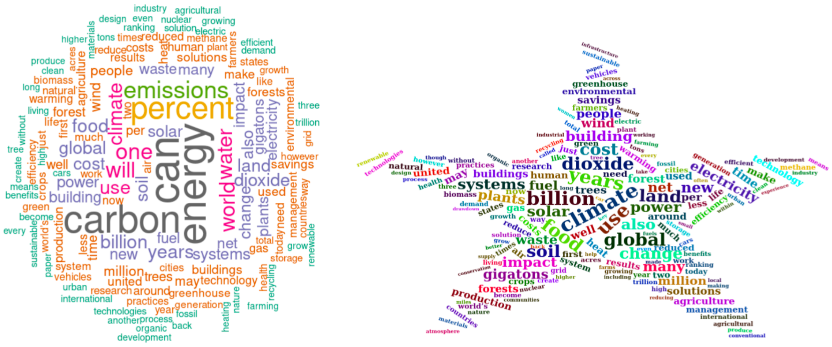
For IOR, we could find similarities to dataset A. This is regarding the initial knowledge that participants can produce code without web browsing. However, it is interesting that the IOR of dataset B can be separated into two groups. One group of participants try to discover answers by increasing the duration of input knowledge time. Another group of participants has low IOR, meaning they are not browsing. Assuming that participants with low IOR and high scores have confident domain knowledge. Also, looking at Figure 3.5 in the DK (domain knowledge) column, it is discovered that CSR, CESR, and IOR have negative correlations. These results imply that a novice programmer's web page access increases. As a result, participants with and without domain knowledge showed similarities in CSR and CESR for each group but not in Score, MBT, VBT, and IOR. Also, a negative correlation was found between domain knowledge and CSR, CESR, and IOR.

RQ2: Do participants with and without domain knowledge each have differences in web browsing and programming activities?

In order to answer this question, we will discuss by looking at Figure 3.5. We found a strong negative correlation between VBT and score for dataset A. Meanwhile, a strong negative correlation can be seen with CSR for dataset B. So, for dataset A, as the score increases, participants tend to search for shorter durations on each web page access. For dataset B, as the score increases, the compile increases or searches decrease. This heatmap presents datasets A and B, each correlating with different features. As a result, we have found differences in participant domain knowledge.

RQ3: Can we estimate whether participants have domain knowledge or not by features calculated from web browsing and programming activities?

Regarding the first research question, CSR and CESR can be used to classify the group of lecture attendees and novice students. Table 3.6 shows the accuracy of binary classification.



(A) Circular shaped word cloud.

(B) Star shaped word cloud.

FIGURE 3.6: Direction of the word clouds prepared for the participant to solve the tasks.

As a result, the Random Forest classification scored the highest mean prediction accuracy of 0.95. Table 3.7 shows the feature importance of the Random Forest classifier. According to the result, the CESR is the most important feature. As a result, participants with lecture attendees and novice students can be predicted.

3.1.4 Experiment 2: Useful Knowledge Extraction

Here, we show findings of useful knowledge extraction [172]. In this experiment, we collected a programming dataset from ten participants. They performed text-mining tasks using the programming language R.

Text-Mining Tasks

In this study, we focused on the creation of word clouds. A simple word cloud results in an image filled with common and perhaps uninformative words known as *stopwords* [147]. A challenge for a programmer unfamiliar with linguistics would be to find the correct domain vocabulary to determine how to remove the stopwords.

Our participants in the case study performed three text-mining tasks using a word corpus prepared by a Txt file. The word cloud is displayed as a Figure 3.6. The prepared word corpus contains 147,103 words and 958,905 characters. The participants were asked to use the R language for programming. The participants solved three tasks (T1, T2, and T3).

- T1: Create a circular word cloud from the corpus like the one in Figure 3.6a.
- T2: Create a word cloud using the corpus without: can, energy, carbon, percent, emissions, one, will, word, or water.
- T3: Create the star-shaped word cloud from the corpus, as shown in Figure 3.6b.

Pre-Post Survey

We asked the participants to complete a pre and post-survey using Google Forms. The pre-survey mainly focuses on understanding the participants' background and programming level. The post-survey mainly focuses on receiving the compiled code and logs exported by TrackThinkTS in CSV format and asking for feedback on the task.

Pre-Survey To understand each participant's background, we ask for the name, age, and gender. To determine the programming level, we ask these questions.

- Q1: On average, how many days per week do you spend on programming?
- Q2: How many years of programming experience do you have?
- Q3: How comfortable are you with programming?
- Q4: How often do you teach/help other programmers?
- Q5: How do you learn new programming skills?
- Q6: How comfortable are you with each programming language?

For Q2 and Q6, we asked them to answer using the self-assessment scale from one to ten. The higher the number in the self-assessment, the better the student is at programming. And for Q6, we asked about the programming languages *C*, *C++*, *Python*, *Scala*, *JavaScript*, *Java*, *HTML, CSS*, *Ruby*, *R* and *Swift*.

Post-Survey For feedback on the assignment, we have asked the following questions.

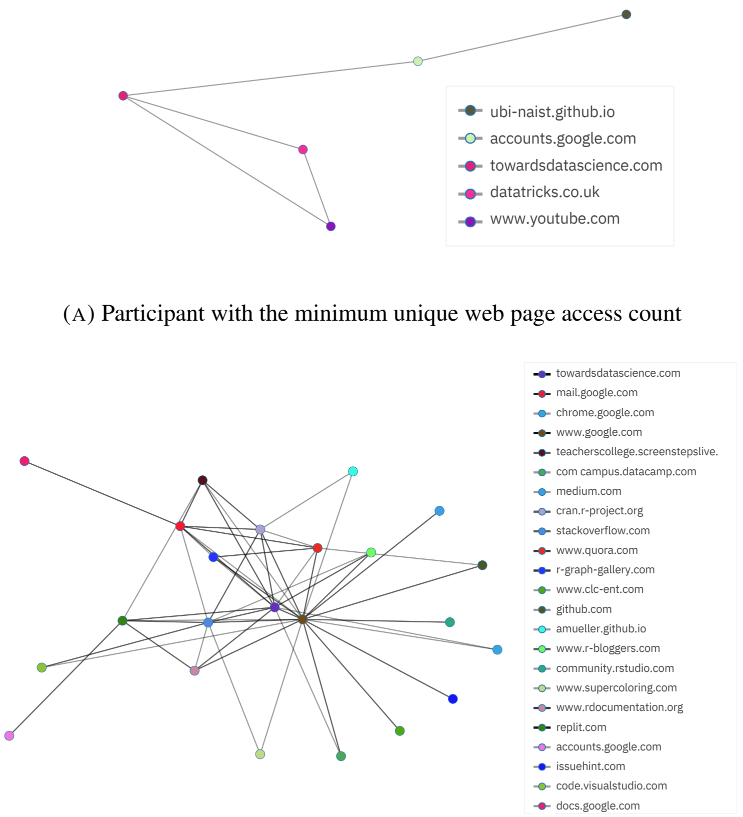
- Q1: How did you feel comfortable during the experiment?
- Q2: Which website supports the answer to T1, T2, and T3?
- Q3: Why was the website useful?

For Q1, we asked them to respond using a self-rating scale of one to ten. The higher the self-rating number, the more satisfied they were with the quality of the task completion. In addition, for Q2, we asked participants to write "none" if they had no websites to support their answers.

Participants & Procedures

We collected data from ten participants (eight males / two females) conducted remotely in the wild, consisting of corporate employees and university students between the ages of 21 and 40 (mean 25.0). The participants had English language skills (one native English speaker) and between four and 14 years of experience with programming languages (mean 6.1). Self-rated programming skills ranged from six to nine out of ten (mean 7.5). These background skills were necessary to ensure that they could perform software programming tasks. Self-rated R language proficiency ranged from zero to five out of ten (mean 2.4). All participants were experienced in programming but unfamiliar with the R language and linguistics.

Before the session began, participants completed a consent form and a pre-survey. We informed all participants that they could withdraw from the experiment and that the collected



(A) Participant with the minimum unique web page access count

(B) Participant with the maximum unique web page access count

FIGURE 3.7: Force-directed network diagram of participants with the minimum and maximum unique web page access count.

logs would be deleted entirely. The consent form included the General Data Protection Regulation (GDPR) consent form. After completing the consent form and pre-survey, each person installs the *TrackThinkTS* [107] extension on their Google Chrome browser. They could only perform a web search within this Google Chrome browser during the study. They will begin solving three text-mining tasks using the R programming language when ready. For the programming editor, we recommend using the VSCode. In this experiment, the time to complete each task is unlimited. However, we designed the experiment to ensure the quality of the work by giving an incentive of a 30 US Dollars Amazon gift card for submitting the correct code. Once the participants are satisfied with the work, we ask them to submit the post-survey, give them the Amazon gift card, and end the experiment.

In this section, we explain the preprocessing of the collected dataset in Section 3.1.4. Then, we explain the visualization techniques in Section 3.1.4.

Preprocessing

Due to the large amount of information TrackThinkTS collects, we try pre-processing the raw data to make it easier to visualize. The first step is to remove the subdirectory from the website logs. For example, if participants are on the website *https://stackoverflow.com/questions/:id*, we make it *https://stackoverflow.com/* in this study. We rephrased the website to use only the domain to keep the visualization manageable for this initial study. In addition to using the domain URL for the visualization, we also create a website category. The website categories are chosen concerning the previous research [93, 105].

- **Official Reference**: Official documents such as *r-project.org*.
- **Technical Blogs**: Not official technical blogs such as medium.com and towardsdatascience.com.
- **Search Engine**: Search engines such as *google.com*.
- **Video References**: Video sources such as *youtube.com*.
- **Q&A Websites**: Question and answer websites such as *stackoverflow.com*.
- **Social Media**: Social media such as *twitter.com* and *facebook.com*.
- **Linguistic Related**: Linguistic websites such as *wordcloud.com*.
- **Not Task Related**: Other uncategorized websites.

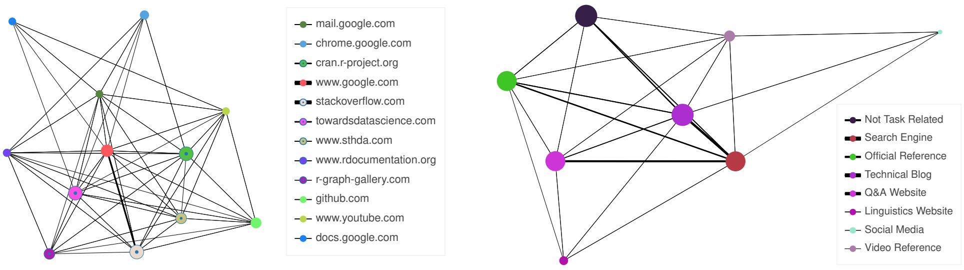
We modify the websites in each category and use them for visualization.

Visualization

For the visualization, we use the force-directed network diagram [15] and the Sankey diagram [139]. We decided to use force-directed network diagrams because this method helps us to understand what kind of website the participants visited. It is a knowledge graph of the participants' search and can easily visualize the web resources they accessed. We used a Sankey diagram to understand the differences in each participant's work. The Sankey diagram shows how long each participant spent on each website or visually categorized. When implementing the Sankey diagram, we only use the time spent accessing the web pages, which is a maximum of 120 minutes. Since the experiment was conducted in the wild, some participants forgot to turn off TrackThinkTS before taking a break. Therefore, there are some gaps in how long the participants looked. This visualization will support further understanding than just having a force-directed network diagram.

Results and Discussions

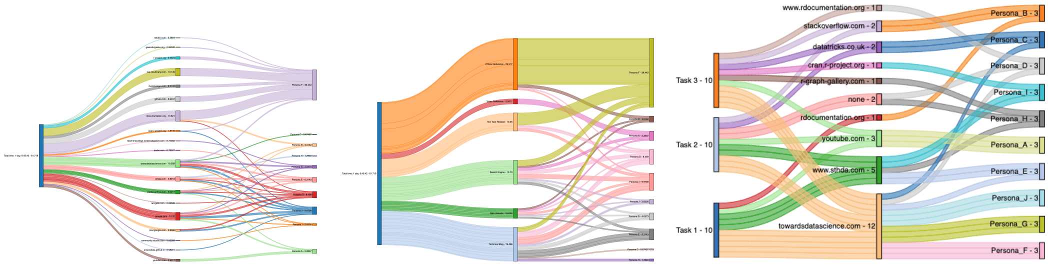
In this section, we show visualized results of the collected data. Figure 3.7 shows the programmer's force-directed network diagrams with the minimum and maximum unique web page access count. The result confirmed that each participant had a different number of nodes to discover the answers through web searches. The participant with the minimal unique web page access count used only five resources, while the participant with the maximum unique count accessed 23 resources. When we look at the other participants' forced-directed diagrams, six out of ten participants searched more than ten web pages for the discovery. The



(A) Diagram of websites accessed by more than five participants. Nodes with dots are the helpful website answered in the survey.

(B) Diagram of all participants data with categorization.

FIGURE 3.8: Force-directed network diagram combining several participants.



(A) Sankey diagram of the minutes spent by users on each website.

(B) Sankey diagram of the minutes spent by users on each category.

(C) Sankey diagram of user website preference from the post-survey.

FIGURE 3.9: Sankey diagram of all the participants.

result confirms that the number of web resources users access is unequal. As a result, it is difficult to identify the valuable website from each participant's search logs.

Figure 3.8 shows the forced-directed network diagram of several participants in this programming study. Figure 3.8a shows the diagram of the websites accessed by more than five participants. The node size indicates that the web resource is being accessed more than the other nodes or web resources. For example, we could say that *cran.r-project.org*, *towardsdatascience.com*, and *stackoverflow.com* is often helpful for the participants to solve the questions. The node with dots in the center represents the websites answered as useful in the post-survey. Thus, the size of the node correlates with its usefulness. On the other hand, websites like *www.rdocumentation.org*, *r-graph-gallery.com*, and *sthda.com* are useful for some people. These are the second or third largest nodes that are potential recommended sites. All of these websites are also selected as supportive in the post-survey. By visualizing the commonly used websites through the force-directed graph, we could determine which websites are widely helpful for the participants. We could also find potential websites that have yet to be discovered that are useful for some candidates. Figure 3.8b shows the force-directed diagram of all participants after categorizing all the websites they visited. The category details are explained in Section 3.1.4. The graph confirms that official websites, technical blogs, and Q&A websites are the main categories for solving R programming problems. It can also be confirmed that the minority in this study uses video references. This characteristic can help us to understand the diversity of the participants in the group.

Figure 3.9 shows the Sankey Diagram of all participants. We defined all ten participants as Persona A through Persona J to describe an individual's behavior. The order of the Persona identification is not significant. Figure 3.9a shows the percentage of access to each web page. It was confirmed that Persona F takes the most time to solve problems, and Persona C takes the least time. Comparing all participants, for Persona A, we found that the participant uses only *youtube.com* as the source. The result is interesting because some participants only used video resources to discover the solution. Regarding time spent viewing the site, *rdocumentation.org* was the site that took the most time to access. We did not see this site from the force-directed network diagram in Figure 3.8a. This result explains that although most participants did not use this site, it may be a potential site for people to use for learning. Figure 3.9b shows how much time each participant spent on each category. The category with the most time spent is in the order of official references, technical blogs, and search engines. The Sankey plot visualizes the percentage and time spent on each web page. Figure 3.9c is a Sankey plot of the website preference collected from the post-survey. We can see that *towardsdatascience.com* was useful for all three tasks. Regarding the result of Figure 3.9a, the site is used more than 10% of the total time spent on the site. We need to ask the participants for a supporting website to visualize the useful website according to the time spent and the number of participants accessing the page.

In summary, the force-directed network and Sankey diagrams support visualizing web searches' individual and group behavior. From individual web browsing behavior, finding a helpful website for solving the task is complex. However, we found that combining the data of multiple programmers can support the discovery of useful websites. Extraction of knowledge from groups of experts has the potential to discover effective web resources.

3.1.5 Conclusion

In this study, we have introduced the TrackThinkTS browser extension. It aims to track users' web search and browsing behavior while emphasizing the importance of maintaining their privacy. We achieve this by using the browser extension API, which allows us to capture events related to tab management, clipboard usage, and web page meta-information. Having access to such data raises privacy concerns. Therefore, we provide an aggregated view of the saved logs for easier management. In addition, users can export the logs and access and manage the raw dataset before submitting them. Giving users extensive control of the log data might alter the dataset's viability. To address this issue, we recommended the participants install different browsers than the ones they usually use in their daily lives and use them to install TrackThinkTS and to proceed with the experiment [6].

In such a case, the users will not need to use the management functionality and remove personal websites and irrelevant URLs. However, even if the students follow this recommendation, there are still a few cases where they need to remove some logs. For example, if a user opens a media website to listen to music while working, he/she can remove the corresponding logs. Recently, TrackThinkTS has been accepted in the Chrome extension store

[6] `https://ubi-naist.github.io/TrackThink/en/usage`

and has been used in experiments involving programming students. The objective is to build a system that provides course supplements to students who need support. The course supplement should provide timely suggestions to the students when they face building errors while solving programming problems. Eventually, the course supplement supplies information and meta-knowledge on how to search for the appropriate solution.

3.2 TrackThink Camera: Tracking Facial Information while Web Browsing

Web search is an essential part of our daily lives. It supports our productivity, creativity, recreation, and even socialization. Previous studies have proposed tools for collecting or visualizing web search logs [28, 107, 120]. These approaches successfully understand users' search activity within a computer or web browser. A study by Aula *et al.* found that most people, novices and experts alike, produce a certain body language when they get stuck in a search [9]. Their research then turned to detecting frustration, or so-called behavioral changes, from the activity inside the computer. However, we can better understand their cognitive/affective states by synchronously capturing their frustration directly from webcam recordings. Therefore, discovering cognitive/affective states from webcam browsing is fascinating.

Estimating cognitive/affective states is a crucial variable affecting human performance in various tasks, including puzzle solving, scuba diving, public speaking, education, fighter aircraft operation, and driving [53]. Understanding the cognitive/affective state during exploration helps us understand how efficient our performance was in finding solutions. The concept of estimating cognitive load in the wild, or Automatic Emotion Recognition (AER) technologies [89], needs to gain attention in research and industry. There is work on estimating the affective state while using a smartphone in the wild [170]. This system lets users get feedback on affective states while using the phone. One of the core techniques for understanding cognitive/affective states is using facial or body information [39, 60, 175].

Determining cognitive states while reading digital textbooks has been done by several researchers [83, 97]. The approaches mainly use an eye tracker or a heat sensor to estimate cognitive states. Previous research has found that body temperature and blink frequency best estimate engagement and can classify users independently as engaged or disengaged [97]. Researchers have also found that pupil diameter and nasal temperature changes correlate with cognitive state [83]. These studies contribute to the understanding of cognitive states during reading in particular. However, all of these researches use additional sensors to detect cognitive states. In addition, these studies focused on something other than reading activity during a web search.

This paper proposes a system that synchronously uses a webcam to collect web search behavior logs and camera-recorded facial and body information. At the same time, a user performs a web search. Figure 3.10 shows the intersection of previous work in the Venn diagram. Our position is to implement an all-in-one technology for tracking web search behavior and synchronizing webcam recordings. The system extends the web search logger TrackThinkTS [107]. This system works with the Google Chrome extension. It does not require any additional sensors or hardware devices. Our main contribution is that we can obtain cognitive/affective information from the user's facial and bodily behaviors without sacrificing this feature. Instead of installing additional sensors or a system to log the video information, anyone can use the system just by installing the Google Chrome extension in the web browser. To the best of our knowledge, we are the first to research and develop an

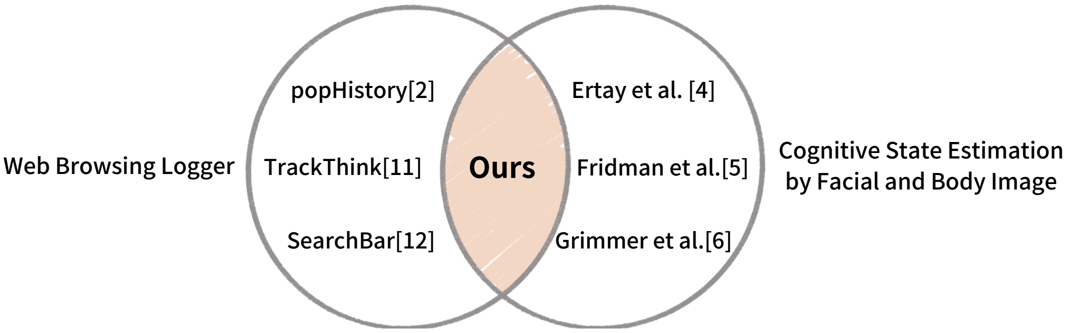


FIGURE 3.10: Position of our proposed system.

all-in-one system to collect web searches and facial and body behaviors.

3.2.1 Architecture

This study will use TrackThinkTS [107] as a baseline system. It collects various information about web search behavior. In this section, we will explain the baseline system in detail. Then, we explain two new specific functions: a recording segmentation function and a webcam recording function. The new system TrackThink Camera operation flow is presented in Figure 3.11.

TrackThinkTS: Baseline System

Makhlouf *et al.* proposed TrackThinkTS as a privacy-aware browsing log tracker. It monitors the following browsing actions and collects logs [107]. First, it collects information about the visited website (e.g., website title, URL, HTML content, viewport width/height, and document width/height). The system keeps a log each time a tab-related operation occurs, such as creating a tab, launching a tab, reloading a tab, or deleting a tab. Second, specific user actions while browsing the website are collected, such as scrolling logs (scrolling speed, scrolling length, and visible text after scrolling is finished) and clipboard logs (the clipboard contents). Compared to typical browsing history, it is possible to collect information such as which parts of the page the user looked at in detail and how long they stayed on each page. Another advantage of TrackThinkTS over other browser loggers is its user-friendly interface for filtering each log. Therefore, study participants using TrackThinkTS can delete privacy-sensitive information before submitting their log files to an experimenter.

Recording Segmentation Function

The traditional TrackThinkTS system collects web activity logs after the extension is installed. Logs are collected continuously until the CSV file is downloaded. Therefore, the existing system records web activity immediately after user installation. To avoid privacy issues when collecting all logs, TrackThinkTS can delete logs before exporting data. In this work, we implement the start and stop buttons to handle the beginning and end of data collection. Therefore, we added a start/stop recording feature to allow users to record web activity

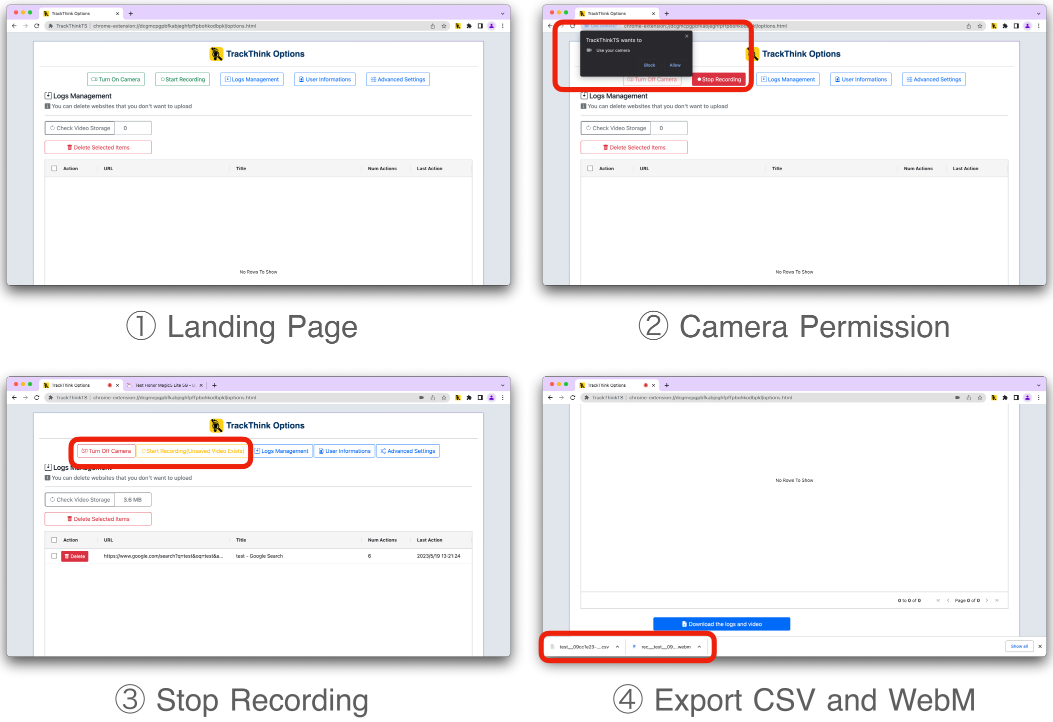


FIGURE 3.11: System Workflow. All-in-one collect web search, facial, and body behavior logs. The system work as a Chrome extension. No additional sensors or devices are required.

data selectively. More precisely, we placed a recording button on the extension's Settings tab to manage the start and stop.

Webcam Recording Function

This section explains the Webcam recording function. Figure 3.11 shows an overview of the system. Each person follows the operation as shown below.

1. **Start Recording** - Click *Turn on Camera* and *Start Recording* buttons for logging.
2. **Allow Recording** - Click *Allow* button for giving permission for camera recordings.
3. **Stop Recording** - Click *Stop Recording* button to stop logging.
4. **Download Logs** - Click *Download the logs and video* button to download files.
5. **Restart Recording** - Click *Start Recording* and logs will be refreshed once.

The new TrackThink Camera allows users to record facial and body recordings using the webcam. Users can also check storage the storage of video recordings while logging. The video can be downloaded locally in WebM format, and a web search logs in CSV. The WebM file name will be generated based on the user ID, name, and recording end time.

3.2.2 Future Work

We want to implement some systems using the TrackThink Camera in future work. One of the ideas is to use appearance-based eye-tracking techniques. We are interested in adding this technology to identify what part of the web page the person was looking at within the web content. This approach will further support understanding the cognitive status of where precisely the user was looking within the web page. Another idea is to quantify the cognitive load of the web page. By doing so, people can choose to read a web page according to its enumerated cognitive load information. Considering GDPR (General Data Protection Regulation), our future work includes secureness on privacy issues. We will implement additional options for exporting data, such as allowing the export of cognitive load estimation results only. Finally, we would also like to create a dashboard for web searchers to retrieve cognitive lifelog data. This way, users can understand which web pages have a high cognitive load. These features will significantly benefit all web search users in the future.

3.2.3 Conclusion

This paper proposes a Google Chrome extension tool, TrackThink Camera, to collect web search activity and facial/body behavior logs. This system supports collecting web search activity and facial/body video recording. Our proposal is the first to collect internal and external information about people while working on web search activity. The new software will be available to anyone with research purposes.

3.3 Appearance based Webcam Eye Tracking

Gaze estimation plays an essential role in many fields. In previous research, it is used to understand reading behavior [79–81], to develop intelligent textbooks for readers [78, 84], and also to analyze confidence or mind wondering while answering questions [21, 82]. The limitation of the above work was that all these studies required specialized hardware to track gaze. In activity behavior computing, replacing the expensive device to estimate the same activity is significant [1].

In gaze estimation, traditional image processing techniques extract features such as pupil position, eye angle, pupil diameter, or gaze direction from eye images. Eye tracking systems, on the other hand, use special cameras or sensors to track eye movements directly. Deep learning has greatly succeeded in various computer vision tasks, including gaze estimation. Gaze estimation has become more convenient with the advent of webcams compared to using skin electrodes in the past [35].

The attached sensor-based method involves sampling the electrical signal from skin electrodes to detect the user's eye movement. The 3D eye model recovery method constructs a geometric model of the eye to determine the direction of gaze. However, it requires the use of special equipment such as infrared cameras.

The 2D feature regression method uses the detected geometric features, such as pupil center and glints, to estimate the gaze direction directly. Like the 3D eye model, the reconstruction method requires using infrared cameras. Funes Mora *et al.* divides eye images into 15 subregions and computes the sum of pixel intensities in each subregion as features [55]. Appearance-based gaze estimation uses the deep neural network to estimate the gaze point. The main difference between conventional appearance-based methods and deep learning-based methods is that the conventional appearance-based method's performance drops when it encounters head motion, while the deep learning method can tolerate head motion. In addition, deep learning methods can extract high-level abstract gaze features from high-dimensional images and learn a highly nonlinear mapping from eye appearance to gaze.

This study aims to estimate gaze position from webcam images. To do so, we create our face dataset using our application. By comparing several methods, we can discover the best prediction model. The evaluation of the model is done by leave-one-participant-out cross-validation. Our contributions are as follows:

1. **Gaze data collection application**: We implement an application for webcam gaze data collection. This application can be used on any laptop.

2. **Top-performing gaze estimation model among our range of models**: We compare several deep learning models to identify the best-performing gaze estimation model.

3. **Discovery of the user dependent features**: We discuss characteristics of high and low prediction rate users from the result of the leave-one-participant-out cross-validation.

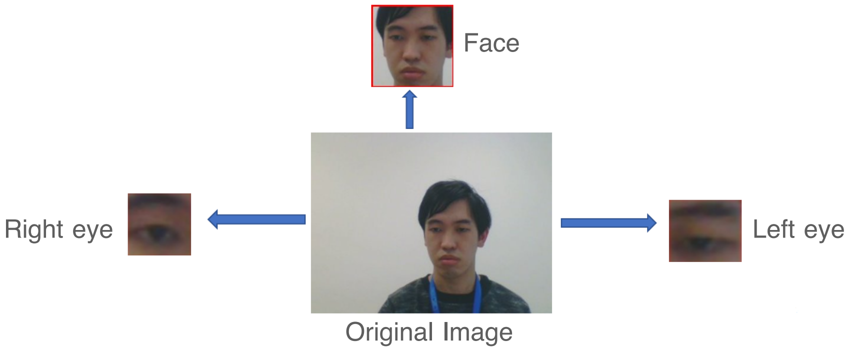


FIGURE 3.12: Image extraction of the right eye and left eye and face.

3.3.1 Architecture

To estimate the gaze points, we used two different methods. The first method uses one feature extractor, and the second uses four different feature extractors.

Data Preparation

Data preprocessing is a crucial stage. Using preprocessing techniques improves the quality and suitability of the image data, making it more suitable for subsequent analysis or processing tasks, such as object detection, image classification, or image segmentation. In addition, directly using the raw gaze images for gaze regression increases computational resources and introduces confounding factors such as scene changes. We cropped some essential parts of the images, such as the face and the left and right eyes, from the original images collected during the experiment for each participant, as shown in Figure 3.12. The images were normalized because normalizing pixel values to a standard range helps to achieve consistency and comparability across different images. In the single feature extractor method, all images (the original images taken during the experiment, faces and left and right eyes) are combined into one image. In contrast, the four feature extractors method uses a single image as input.

Deep Learning Model

Three different backbones were used to compute the gaze points: VGG16, ResNet50, EfficientNetB2, and EfficientNetB7. Different combinations with the backbones were used in terms of image resolution (64 × 64, 128 × 128, 256 × 256), batch size (8, 16, 32), number of trainable layers in the backbone (all, last layer, last two layers, none), and backbones with the same weights as ImageNet or without the same weights as ImageNet (i.e. training from scratch) We incorporated a learning rate of 5e-5 alongside the ReduceLROnPlateau callback. This callback plays a vital role by automatically adjusting the learning rate during training and continuously monitoring a specified metric like validation loss. If the monitored metric shows no further improvement, the callback reduces the learning rate accordingly. Additionally, we utilized the “*Adam*” optimizer for our model optimization. The process we undertook can be regarded as an experiment. We explored various combinations to determine the most effective approach for gaze estimation. We aimed to thoroughly investigate

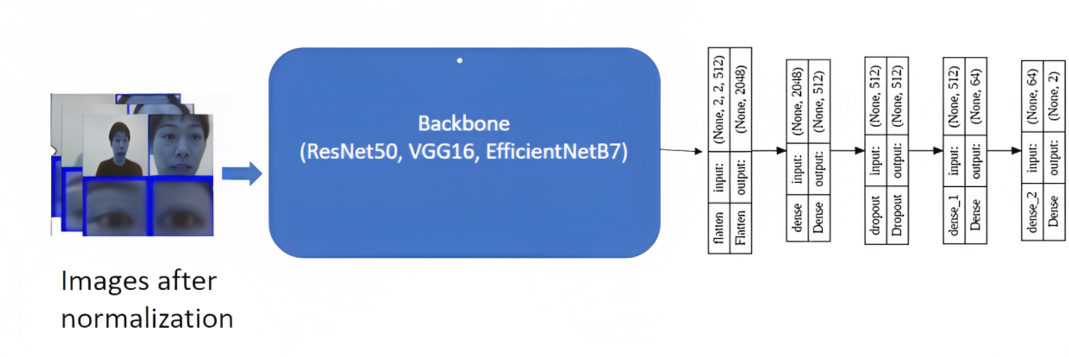

Images after normalization
Backbone
(ResNet50, VGG16, EfficientNetB7)
flatten input: (None, 2, 2, 512)
Flatten output: (None, 2048)
dense input: (None, 2048)
Dense output: (None, 512)
dropout input: (None, 512)
Dropout output: (None, 512)
dense_1 input: (None, 512)
Dense output: (None, 64)
dense_2 input: (None, 64)
Dense output: (None, 2)

FIGURE 3.13: Architecture of the method which uses one feature extractor.

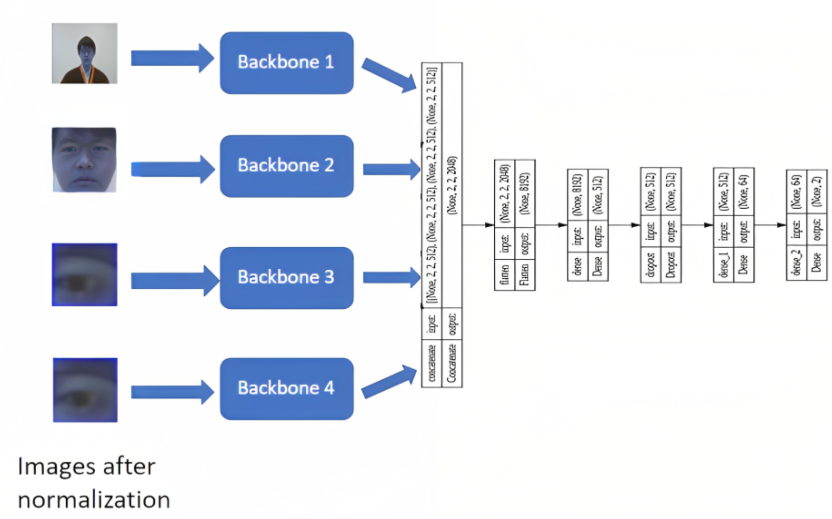

Backbone 1
Backbone 2
Backbone 3
Backbone 4
concatenate input: [(None, 2, 2, 512), (None, 2, 2, 512), (None, 2, 2, 512), (None, 2, 2, 512)]
Concatenate output: (None, 2, 2, 2048)
flatten input: (None, 2, 2, 2048)
Flatten output: (None, 8192)
dense input: (None, 8192)
Dense output: (None, 512)
dropout input: (None, 512)
Dropout output: (None, 512)
dense_1 input: (None, 512)
Dense output: (None, 64)
dense_2 input: (None, 64)
Dense output: (None, 2)
Images after normalization

FIGURE 3.14: Architecture of the method which uses four feature extractors.

and compare different combinations of models, considering factors such as top-1 accuracy, top-5 accuracy, the number of parameters, and model depth [36]. By conducting this comprehensive analysis, we sought to identify the combination that would yield the best results for gaze estimation.

The method using a single feature extractor takes a single input constructed by combining the original image, the face, and the left and right eyes. Then, this input is fed into a backbone (VGG16, ResNet50, EfficientNetB7). In our selection process, we carefully considered the performance metrics of top-1 accuracy and top-5 accuracy when choosing the VGG16, ResNet50, EfficientNetB7, and EfficientNetB2 models. We aimed to assess whether models with a higher or lower number of parameters yielded better results. Additionally, we considered the depth of the models to ensure a comprehensive evaluation of their capabilities. Then, the feature maps are passed to the fully connected layer, and finally, the network outputs the gaze pixel coordinates as shown in Figure 3.13.

$$pixel\ coordinates \in \mathbb{R}^2$$

TABLE 3.8: The best results using one feature extractor method.

Model	Image Resolution	Batch Size	Trainable layer of Backbone	RMSE (px)	RMSE (cm)
EfficientNetB7	64×64	32	All	271.490	4.848
EfficientNetB7	64×64	32	Last	262.567	4.688
ResNet50	64×64	8	All	152.236	2.718
VGG16	64×64	32	All	147.168	2.628
ResNet50	64×64	16	All	146.577	2.617
VGG16	64×64	16	All	139.425	**2.489**

TABLE 3.9: Comparison of best results using four feature extraction method.

Model	Resolution	Batch Size	Trainable layer of Backbone	RMSE (px)	RMSE (cm)
EfficientNetB2	64×64	32	All	213.906	3.819
ResNet50	64×64	32	All	141.319	2.523
VGG16	64×64	32	Last two	134.419	**2.400**

In contrast, the method using the four feature extractors takes four different images as input: the original image, the face, and the left and right eyes. Then, these four images are fed to four different backbones, and the feature maps from each backbone are passed through a concatenation layer, where they are concatenated. Then, the concatenated feature maps are passed to the fully connected layer, and finally, the network outputs the gaze pixel coordinates as shown in Figure 3.14.

$$pixel\, coordinates \in \mathbb{R}^2$$

Model Accuracy Comparison

We were interested in determining each participant's individual contribution to the overall result. To assess each participant's impact, we used a technique known as leave-one-participant-out cross-validation. In this method, one participant is excluded from the training set and used as the test set, while the remaining participants are included in the training set.

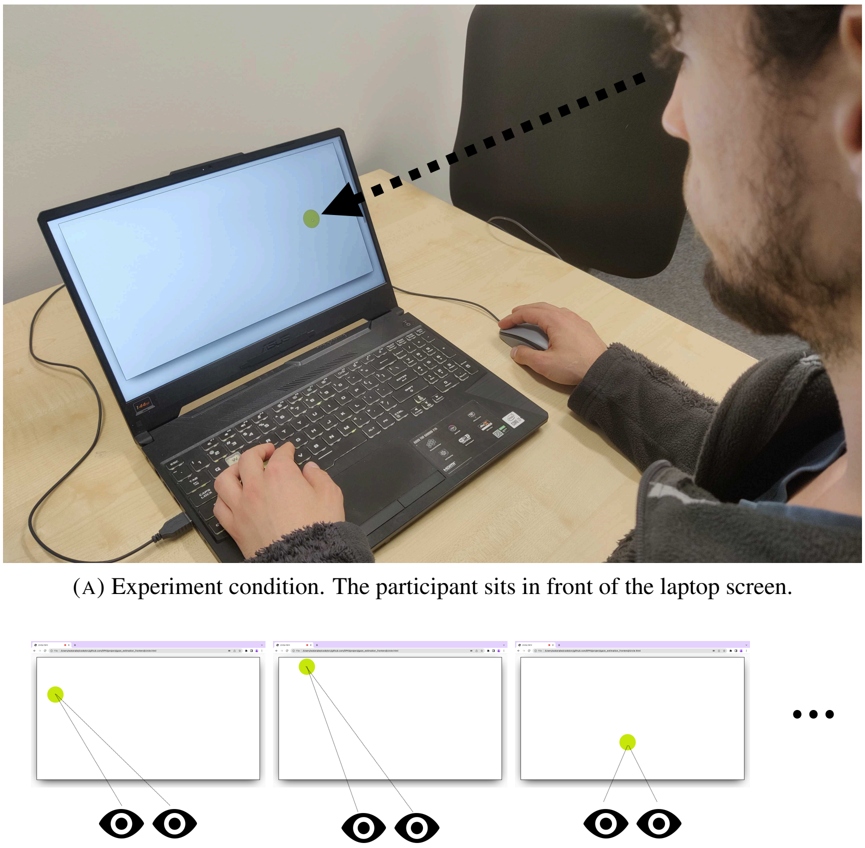
We used the root mean square error matrix to evaluate the error difference between the ground truth gaze points and the predicted gaze points. Note that we used pixel coordinates and computed the error difference in centimeters. In order to convert the pixel error difference to the centimeter error difference, we made some simple calculations. The following calculations refer to PPC: Pixels Per Centimeter, SWR: Screen Width Resolution, SWL: Screen Width Length, and RMSE: Root Mean Square Deviation, respectively. SWR and SWL are our experimental screen-dependent variables, as explained previously in Figure 3.16.

$$PPC = SWR/SWL = 1920(px) \div 34.5(cm) = 55.65 \approx 56(px/cm)$$

$$RMSE(cm) = RMSE(px) \div PPC = RMSE(px) \div 56(px/cm)$$

3.3.2 Experimental Design

This Section explains the process of collecting the face image and the laptop screen position data. Figure 3.15 shows an overview of the experimental settings and the data collection



(A) Experiment condition. The participant sits in front of the laptop screen.

(B) Experiment work-flow. A participant looks at the circle on the screen and clicks with the mouse cursor.

FIGURE 3.15: Data collection experimental setting and the workflow.

workflow. We will explain the background information about the participants in Section 3.3.3 and the data collection procedure in Section 3.3.4.

3.3.3 Participants

Our experiment collected data from 17 participants (12 males and five females). Along with the gaze points, we recorded their background information, such as their country of origin and whether they had to wear glasses during the experiment. Of the 17 participants, nine were from Japan, five were from India, and the rest were from Hungary, Chile, and Morocco. Before the experiment, we obtained consent from the participants regarding the General Data Protection Regulation (GDPR). The participants were allowed to opt out of the experiment at any time. At the end of the experiment, all participants who completed it received a ten-euro Amazon voucher.

3.3.4 Data Collection Procedure

In this study, we experimented using a single laptop computer. The experiment was conducted in the same room in a controlled manner. Figure 3.15 shows the data collection experimental setting. The data collection was done using the following procedure:

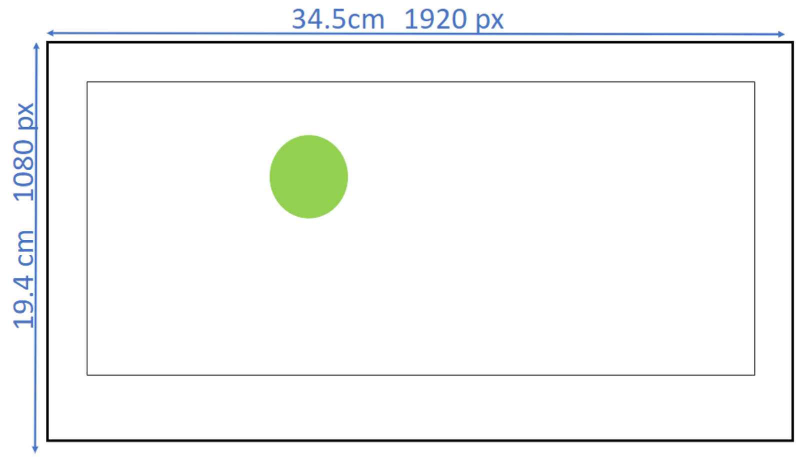


FIGURE 3.16: Experiment dimensions.

1. Participants were positioned at an approximate distance of 30 cm from the webcam.
2. The experiment conductor explains the process and the purpose of data collection.
3. Fill out the agreement on the consent form.
4. Sit in front of the laptop and direct their attention towards the circle on the screen.
5. Click on the circle using the mouse cursor. The camera captured an image and recorded the pixel coordinates corresponding to the click location.
6. The circle will randomly move to another position on the screen.
7. Repeat Steps 4-6, and the process will end when 50 images are saved.

The data we collect are the pixel coordinates of the laptop screen and facial images associated with each clicked circle. In total, 50 sets of pixel coordinates and face images were stored for 17 participants, and 850 sets of pixel coordinates and face images were collected. Figure 3.16 shows the dimensions of an experiment laptop. The screen resolution was $1080px \times 1920px$ with a width of $19.4cm \times 34.5cm$.

The experiment was conducted in an approximately 15-minute session in a controlled environment within a closed, empty room. This approach ensured consistent lighting conditions and helped minimize any potential background noise so as not to interfere with the gaze data. The laptop was placed in a stable position.

3.3.5 Results

In this Section, we explain the result of comparing each deep learning model and leave-one-participant-out cross-validation.

Table 3.8 shows the result using one feature extractor from the different combinations of settings, and it presents the best two results from each of the backbone or feature extractors. We found VGG16 with image resolution 64 × 64, batch size 16, and all trainable layers with the same weight as imagined produce the best result with an error difference of 2.489 cm.

TABLE 3.10: The result of Leave-One-Participant-Out cross-validation.

User ID	Gender	Glasses	RMSE (px)	RMSE (cm)
P1	Male	Yes	150.066	2.679
P2	Male	No	150.339	2.684
P3	Male	No	149.632	2.672
P4	Male	No	148.674	2.654
P5	Male	No	148.029	2.643
P6	Female	No	145.878	2.604
P7	Female	No	149.390	2.667
P8	Female	Yes	141.857	**2.533**
P9	Male	No	266.543	**4.759**
P10	Male	No	259.303	4.630
P11	Male	Yes	257.719	4.602
P12	Male	No	251.260	4.486
P13	Female	No	249.370	4.453
P14	Male	Yes	230.259	4.111
P15	Male	No	190.853	3.408
P16	Female	Yes	165.613	2.957
P17	Male	No	159.160	2.842
Mean			189.056	3.375
Standard Deviation			49.911	0.891

Table 3.9 shows the result for each backbone with the best setting using the four-feature extractors method. It can be seen that the VGG16 with the image resolution 64 × 64, batch size 32, and only the last two trainable slices with the same weight as the ImageNet setting again outperformed the others and gave the best result with an error difference of 2.400 cm.

Table 3.10 summarizes the cross-validation result where one participant is evaluated based on the remaining participants. In this method, we have used the same setting as the best method with four feature extractors, i.e., VGG16 backbone with image resolution 64 × 64, batch size 32, and only the last two trainable layers with the same weight as the ImageNet. The lowest error difference is 2.533 cm, and the highest error difference is 4.759 cm among the participants.

Regarding these results, we found that VGG16 performs well for appearance-based gaze estimation. Using VGG16 to perform leave-one-participant-out cross-validation, we got a mean error of 3.375 ± 0.891 cm.

3.3.6 Discussion

Regarding the experiment, the performance of multi-feature extractors was observed to surpass that of methods employing single-feature extractors. Each feature extractor is designed to capture specific information from the input data. We can leverage the diverse information they offer by combining multiple feature extractors. Each extractor can focus on distinct aspects or patterns in the data, resulting in a more comprehensive representation. By extracting features from multiple extractors and merging the outputs using fusion techniques such as averaging, concatenation, or advanced methods like attention mechanisms, the overall performance can be enhanced, leading to a more robust representation. Contrary to the initial assumption, the results indicated no significant difference in error rates between individuals who wore glasses and those who did not. This finding is intriguing as it demonstrates the broad applicability of our model across various users, including individuals who wear glasses. Individual differences among participants, such as eye shape, size, and movement patterns, can affect gaze estimation accuracy. Factors like fatigue or blinking frequency can also introduce variability in the estimation results. By delving into these aspects in future research, we can better understand the factors influencing error differences in participants' gaze estimations using webcams. This knowledge can contribute to developing more accurate and robust gaze estimation methods in various applications.

For future work, there are several aspects we aim to address. Firstly, we intend to enhance the model's robustness by collecting additional data from diverse backgrounds, including variations in room lighting and zoomed-in and zoomed-out images. Secondly, the VGG16 model will be evaluated on publicly available datasets. A robust model for appearance-based eye tracking is important for the next task. Thirdly, expanding the participant pool and gathering data from more individuals is another task we plan to undertake. Fourthly, instead of random circles appearing on the screen, we can modify the experiment by controlling the number of circles in each quadrant to ensure data balance and mitigate potential issues. Lastly, an important aspect of our future endeavors involves designing publicly available gaze prediction software.

Additionally, there is potential to explore enhancements in the computational and memory costs associated with the method that employs four feature extractors in the future. Furthermore, gaze estimation can be done using transformers in the future. By leveraging transformer-based models, we can potentially improve the accuracy and performance of gaze estimation in our system.

3.3.7 Conclusion

This research presents an approach to collect data for modeling webcam gaze estimation based on appearance. A total of 17 participants were involved, and we collected 50 patterns of face images along with corresponding pixel coordinate information. The findings reveal that utilizing a VGG16 backbone with four feature extractors yields the most accurate results for gaze estimation. Through leave-one-participant-out cross-validation analysis, we observed that the participants' root mean square deviation ranged from 2.533 cm to 4.759 cm, with an average error value of $3.375 \pm 0.891 cm$. Our future work will expand our data collection efforts to encompass diverse datasets. This expansion aims to enhance the robustness of our model for gaze estimation.

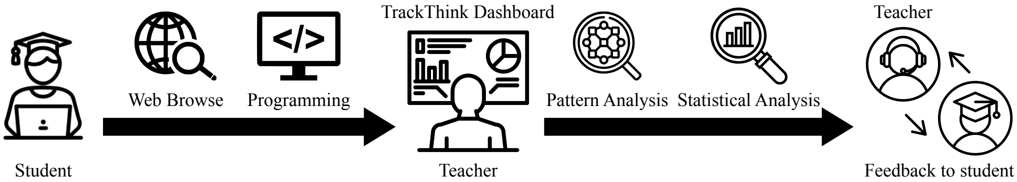


FIGURE 3.17: Entire process of using TrackThink Dashboard.

3.4 TrackThink Dashboard: Understanding Self-Regulated Learning in Programming

In recent years, programming education has received significant attention as an essential skill for working professionals. To meet the demands of a technology-driven world, it becomes imperative for educators to facilitate hands-on learning experiences that foster students' self-regulated learning (SRL) skills [191]. SRL has been characterized by learners taking control of the learning process through goal setting, monitoring, and reflection to improve academic performance and long-term knowledge retention. Society needs to improve students' ability to use the Internet to solve complex problems such as search engines or chat-based platforms ChatGPT [129] to find appropriate information. In order to educate, taking control of understanding students' choice decisions is significant.

To support the development of SRL skills in programming education, we present TrackThink Dashboard, an innovative application designed to visualize the SRL process. The primary goal of the TrackThink Dashboard is to empower both students and teachers in programming education. The system integrates web browsing and programming activities, providing students with a holistic view of their learning journey and enabling them to make informed decisions about their learning strategies. Students can use the application to track their decision-making workflow and visualize their choice of information over time. By visualizing the connections between web browsing and programming activities, students gain a deeper understanding of why they choose web resources. This application allows students to develop meta-cognitive skills that recognize gaps in their knowledge, seek additional resources, and adjust their learning strategies accordingly. For teachers, TrackThink Dashboard offers a comprehensive dashboard that provides insights into student decision-making factors, study patterns, and areas of difficulty. The application enables teachers to identify struggling students and provide targeted interventions to support their learning journey. In addition, the platform facilitates communication and collaboration between teachers and students, enhancing the educational experience.

In previous studies, collecting logs of web browsing and programming has been done. For collecting browsing logs, there are systems such as SearchBar [120], popHistory [28], and TrackThinkTS [107]. For collecting programming logs, there are systems like Log++ [108], Projection Boxes [104], and Log-it [86]. However, previous research needed to work on visualizing the knowledge acquisition process (web browsing) and the knowledge output process (programming) together as a flow. We aim to visualize the workflow of web browsing and programming.

In this study, we propose an application that anyone can use to visualize the self-regulated learning workflow of students' programming studies. Figure 3.17 shows the overall process. We aim to achieve this by using web browsing activity logger [107] and web programming Integrated Development Environment (IDE) [77] for collecting logs. We propose the system TrackThink Dashboard, which visualizes the synchronous workflow of web browsing and programming. To allow non-programmers to use it, we implemented the GUI (Graphical User Interface) dashboard, which is easy to use. We attach the log files to visualize the web browsing and programming logs as a flow chart. Our contributions to this paper are the following:

1. **Propose a system to visualize web browsing and programming in a single chart**: We are the first to visualize web browsing and programming flow into single chart. To the best of our knowledge, none of the previous work tackle to visualize two activities into single flow-chart.
2. **Discover web browsing and programming patterns for self-regulated learning**: Using our visualization tool, we discover patterns of how student use web browsing to seek information and output as programming.

3.4.1 Architecture

We use two applications for web browsing and programming activity logging to visualize web browsing and programming workflows synchronously. Figure 3.18 shows an overview of our system architecture. This section explains data source, fusion, and visualization approaches for understanding self-regulated learning.

Data Source

In this section, we explain the system's data source. Our architecture allows adding of more sensors or software application logs. In our prior study, we aim to use web browsing and programming logs. To collect web browsing activity logs, we use TrackThinkTS [107], the web browser extension. We use C2Room [77], an online programming IDE for collecting programming activity logs. We will explain web browsing and programming logs in detail.

Web Browsing Logger

Web browsing logs are recorded through TrackThinkTS [107], explained in Section 3.1. One notable advantage of TrackThinkTS over other web browser loggers is its user-friendly interface, which allows participants to quickly delete irrelevant logs collected during the experiment. For a detailed overview of the logs collected by TrackThinkTS, refer to Table 3.11. Users need to install this extension in their web browser to utilize it. In our study, we have selected Google Chrome [7] as the preferred web browser.

[7] `https://www.google.com/chrome/`

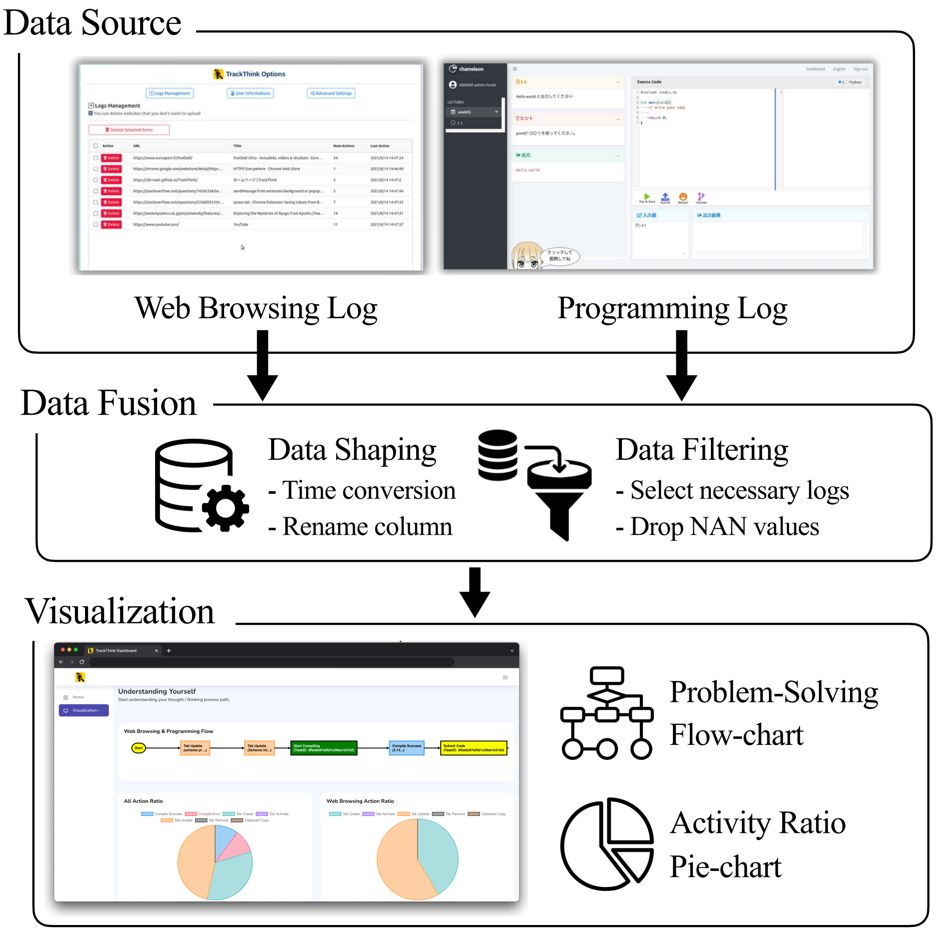


FIGURE 3.18: System Architecture. Data sources are reshaped and filtered prior to visualization. The visualization shows flow-chart and pie-chart.

Programming Logger

Programming activities are recorded through an online web IDE called C2Room [77]. This system is specifically tailored for online programming classes in educational institutions and companies. For a detailed overview of the logs collected by C2Room, refer to Table 3.12. Our study focuses on using specific user programming actions derived from these logs. By analyzing these actions, we gain insights into how and when users execute their code in the compiler and the corresponding feedback received for each task. This analysis extends even to instances where users are satisfied with their written code and proceed to submit it.

Data Fusion

Data fusion is applied to logs collected from the data source. This data fusion process comprises two primary procedures: data shaping and filtering. Data shaping involves transforming raw data into a unified format suitable for combining various data sources. On the other hand, data filtering entails selecting relevant data points following the conversion and concatenation. These processes play a crucial role in simplifying visualization, enabling students

TABLE 3.11: Detail of the web browsing log collected by TrackThinkTS.

Column Name	Description
UserID	User ID of the participant who experimented.
UserAction	User action category. Tab, scroll, and clipboard copy actions.
date	The timestamp of the log collected. It is collected in UNIXTIME.
Tab_URL	The URL of the web page accessed.
Tab_Title	The title of the web page accessed.
Tab_BodyText	The body information of the web page accessed.
ClipboardCopy	The selected text of clipboard copy.
Scroll_YAxisSpeed	The speed of the vertical scroll.
Scroll_VisibleText	The text visible for user after the scroll stop.
Scroll_ViewPort_XAxisScroll	The viewport of the horizontal scroll.
Scroll_ViewPort_YAxisScroll	The viewport of the vertical scroll.
Scroll_ViewPort_XAxisScrollRate	The percentage of the horizontal scroll.
Scroll_ViewPort_YAxisScrollRate	The percentage of the vertical scroll.
Scroll_ViewPort_ViewPortWidth	The length of the width of a viewport.
Scroll_ViewPort_ViewPortHeight	The length of the hight of a viewport.
Scroll_ViewPort_DocumentWidth	The length of the width of a document.
Scroll_ViewPort_DocumentHeight	The length of the height of a document.

TABLE 3.12: Detail of the programming log collected by C2Room.

Column Name	Description
time	The timestamp of the log collected. It is collected in JST.
uid	User ID of the participant who experimented.
classID	Class ID of the virtual room. The session organizer creates it.
taskID	Task ID of the question. The session organizer creates it.
lang	Programming language selected to compile or submit.
op	The user operation category. Such as compiling or submitting code actions.
msg	The message after the compile. Such as response status, output, and error message.

TABLE 3.13: Result of data fusion. Selected log after data shaping and filtering.

Column Name	Description
timestamp	The timestamp when the action has been occurred. The unit is in UNIXTIME.
userID	Participant ID while conducting a self-regulated learning.
taskID	Task ID of the question. The session organizer creates it.
userAction	All action logs collected by data resource.
tabURL	The URL of the web page accessed.
clipboardCopy	The selected text of clipboard copy.
msg	The message after the compile. Such as response status, output, and error message.

TABLE 3.14: Detail of the elements inside user action after data fusion.

User Action	Description
Tab creation	Opening a new tab.
Tab activation	Selecting an existing tab.
Tab refresh	Reload a web page or open a new page in the same tab.
Tab remove	Delete a ta.b
Clipboard copy	Clipboard copy action.
Start compiling	Action of user start compiling code.
End compiling	Action of compiling end.
Submit code	Action of user submitting code.

and teachers to comprehend the performance metrics easily during self-regulated learning in programming. We will now provide a detailed explanation of each procedure.

Data Shaping

We employ data shaping as a preprocessing step to facilitate the concatenation and filtering of logs. Firstly, we rename the columns for each web browsing and programming log. Specifically, we begin by renaming the columns *date* and *time* to *timestamp*. Additionally, we rename *UserID* and *uid* to *userID*. For *UserAction* and *op*, we change to *userAction*. This column renaming process aims to align the corresponding information between the two applications. Once the column renaming is complete, we convert the *timestamp* values to UNIXTIME and sort them chronologically.

Data Filtering

Table 3.13 shows the table after data filtering. Some information is removed, such as window scroll speed from the web browsing log, class ID from the programming log, or logs involving NAN values. In order to sort logs into time order, we convert all units of the timestamp into UNIXTIME. Specifically, the programming logs collected by C2Room were converted from JST to UNIXTIME. All the logs are concatenated and sorted in a time order using *timestamp*. Table 3.14 shows the elements of *userAction* after concatenating the web browsing and programming logs.

3.4.2 Visualization

Visualization of web browsing and programming activity is performed after data fusion. Figure 3.19 shows the library used for visualization. For the visualization, we chose a flow chart and pie chart as an approach. We will explain the implementation process and the reason for the choice in detail.

The flow-chart is selected to visualize the problem-solving progress in a time series. The approach uses a flow chart to understand what search results participants used to arrive at their answers and what compilation errors they encountered when re-running their searches.

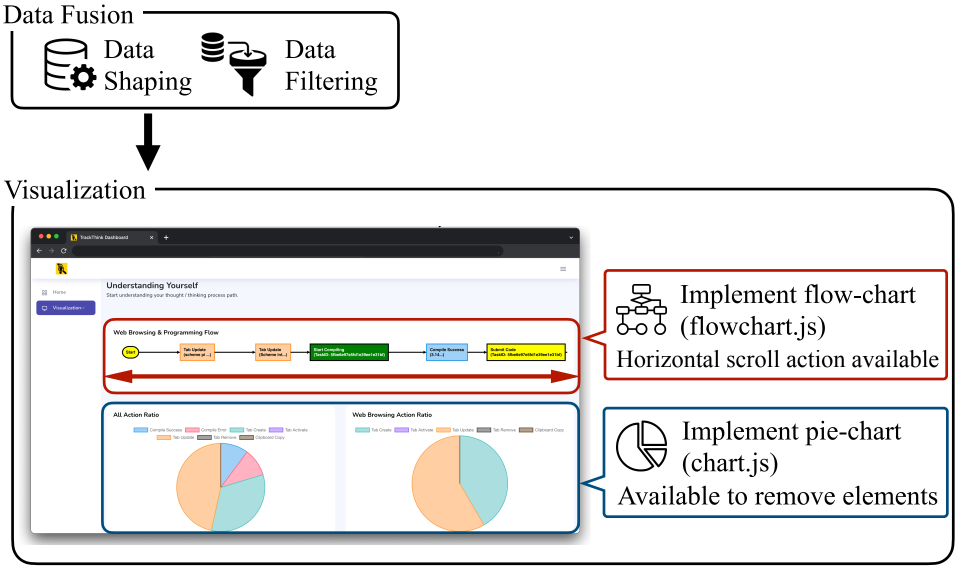

Data Fusion
Data Shaping
Data Filtering
Visualization
TrackThink Dashboard
Home
Visualization
Understanding Yourself
Start understanding your thought / thinking process path.
Web Browsing & Programming Flow
Start
Tab Update (scheme pi ...)
Tab Update (Scheme int...)
Start Compiling (TaskID: 5fbe6e97e5fd1e39ee1e31bf)
Compile Success (3.14...)
Submit Code (TaskID: 5fbe6e97e5fd1e39ee1e31bf)
All Action Ratio
Compile Success
Compile Error
Tab Create
Tab Activate
Tab Update
Tab Remove
Clipboard Copy
Web Browsing Action Ratio
Tab Create
Tab Activate
Tab Update
Tab Remove
Clipboard Copy
Implement flow-chart (flowchart.js)
Horizontal scroll action available
Implement pie-chart (chart.js)
Available to remove elements

FIGURE 3.19: Visualization of the data into flow-chart and pie-chart.

The flow-chart is implemented using *flow-chart.js* [135]. It is a JavaScript library for flow-chart SVG (Scalable Vector Graphics) rendering that runs in the terminal and browser. We categorize users' activities in different colors and shapes for start and stop edges. The work-flow is not visualized fully on the screen, but the user can scroll horizontally to see the actions between the start and end edges. For edges like *tab activate* or *tab update*, the hyperlink is set so that users can jump to the webpage once they tap the edge.

The pie-chart is selected for visualization because the activity ratio of user action is essential in identifying users' domain knowledge [174]. The pie chart is implemented using *chart.js* [41]. Clicking each element removes a specific activity from the pie chart. The option of removing elements helps teachers or students to focus on the ratio of an activity that they want to compare. It is a JavaScript library for making HTML-based charts.

The uniqueness of our proposed application is that it is made in the form of a GUI. The application automatically visualizes once the user inserts the collected CSV files into the TrackThink Dashboard. Non-programmers like teachers in school can easily play the operation.

3.4.3 Experimental Design

In this section, we explain the process of data collection. First, we explain the participants' background and number in detail. Then, we explain the details of the experiment procedure and how we asked the participants to work on the experiment.

Participants

In this experiment, we collect logs from lecture students (Dataset A) and non-lecture students (Dataset B). Non-lecture students still need to take the university programming course. The lecture students are students who learned about programming in a university lecture. The total number of participants is 33 unique (32 males and one female) university students in Japan.

Dataset A – Group of university students attending lectures We collect data from 13 unique university students (12 males and one female) in Japan. Participants have taken university courses in the Scheme programming language. Therefore, participants have some knowledge from the class, such as Scheme grammar or syntax.

Dataset B – Group of university students not attending lectures We collect data from 20 unique (20 males) university students in Japan. Participants did not take any university courses related to the Scheme. Therefore, participants do not have any knowledge from the class, such as Scheme grammar or syntax.

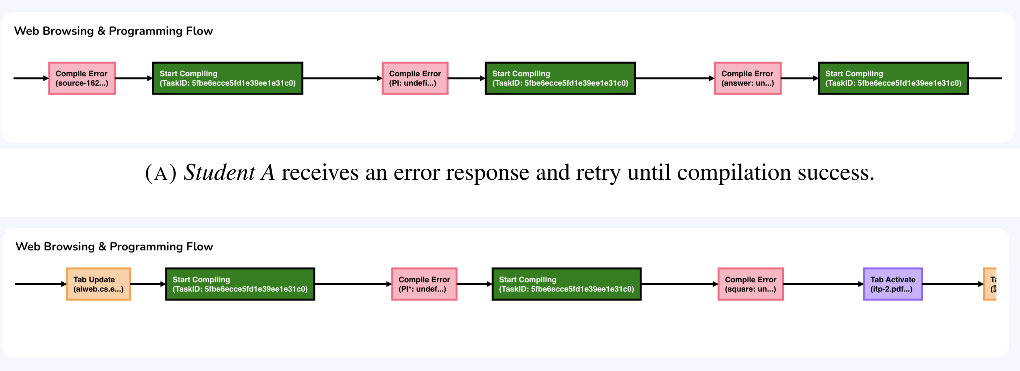
Experiment Procedure

Figure 3.3 shows the condition of the experiment. Before the experiment, C2Room and TrackThinkTS were installed on each participant's laptop. Under these conditions, the experiment was conducted using the following procedure. First, the experiment conductor presents the purpose of the experiment, experimental conditions, and tools that will be provided to the participant. Only the person who agrees with the conditions can participate in this experiment. Second, the participant enters the personal workstation, and the web browsing and programming loggers begin recording data. Third, participants worked on solving problems with a given schema using the C2Room programming editor. The order of question-solving is not restricted, but it is assumed that the participant will solve the easy questions step by step. C2Room will record the compiled results for each question. Fourth, participants may use a web search engine to find the answer. TrackThinkTS will track web browsing behavior. Fifth, the participant submits the answer and moves on to the next question when satisfied with the code compilation result. Allow the student to return to previous questions to change the answer. Sixth, when all questions have been answered or an hour has passed, we stop the participants from working on problem-solving. Last, participants remove privacy-sensitive logs from TrackThinkTS. Once all recorded logs are submitted to the experimenter, the participant leaves the personal workspace.

Table 3.5 shows the questions asked to answer. We have prepared ten questions in order of difficulty. Easy questions are those that require fewer lines of code to solve. In this experiment, we chose Scheme (Racket), one of the dialects of LISP languages, as the programming language of the task [2]. There are the following reasons for choosing it.

1. It is used in programming courses at several universities because of its simple language specification.

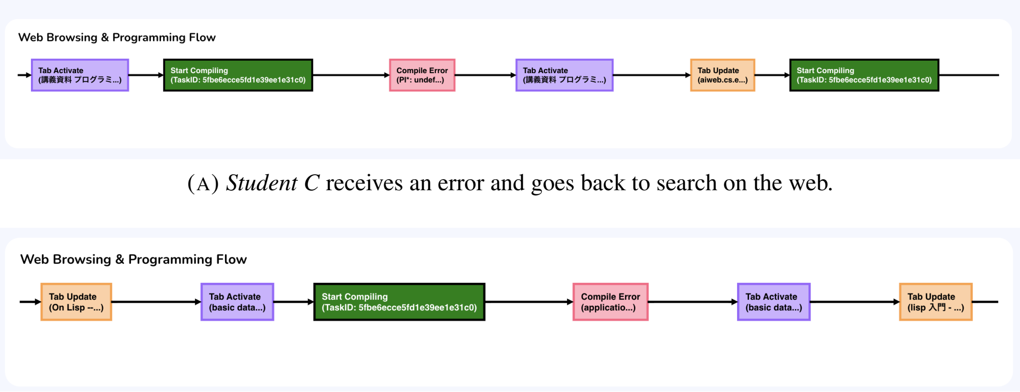
2. It is tailored for lecture use, so its language specification is usually unknown to students except those attending the lecture.


Web Browsing & Programming Flow
Compile Error (source-162...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Compile Error (PI: undefi...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Compile Error (answer: un...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Web Browsing & Programming Flow
Tab Update (aiweb.cs.e...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Compile Error (PI*: undef...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Compile Error (square: un...)
Tab Activate (itp-2.pdf...)

(A) *Student A* receives an error response and retry until compilation success.

(B) *Student B* receives an error response and retries compile once more.

FIGURE 3.20: Group of try-and-error students. Students receive an error response after compiling, and students try to compile before going back to the web search.


Web Browsing & Programming Flow
Tab Activate (講義資料 プログラミ...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Compile Error (PI*: undef...)
Tab Activate (講義資料 プログラミ...)
Tab Update (aiweb.cs.e...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Web Browsing & Programming Flow
Tab Update (On Lisp --...)
Tab Activate (basic data...)
Start Compiling (TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Compile Error (applicatio...)
Tab Activate (basic data...)
Tab Update (lisp 入門 - ...)

(A) *Student C* receives an error and goes back to search on the web.

(B) *Student D* receives an error and goes back to search on the web.

FIGURE 3.21: Group of try-and-search students. Students receive an error response after compiling, and students return to the web search activity to find a solution before the following compilation.

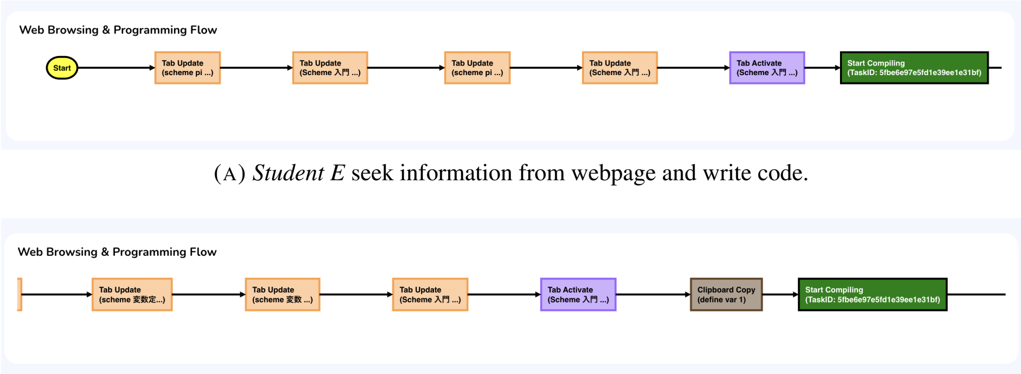
Questions are easy for lecture attendees. Questions are difficult for non-lecture attendees to solve from scratch, requiring web searches to understand basic Scheme syntax or grammar rules. The question level can capture a variety of student problem-solving behaviors.

3.4.4 Results

In this section, we show results obtained from the TrackThink Dashboard. Student search queries often show the Japanese language due to the data collection of university students' backgrounds.

Figure 3.20 shows a sample student workflow for solving a programming problem using the try-and-error approach. Students compile the code, and after receiving an error response, they modify the code and try to recompile it. Some students do a try-and-error flow more than twice before returning to the web browser to find an answer.

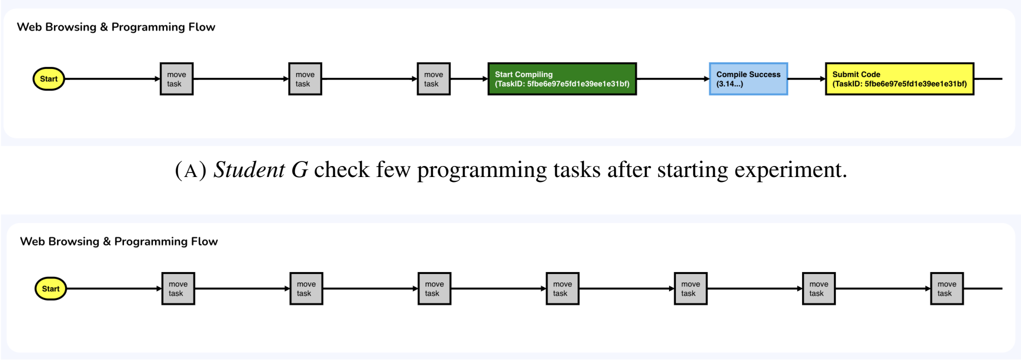
Figure 3.21 shows an example student working on the same task and receiving an error response, but this user decides to go back to the web search before the following compilation. We call this pattern of solving a try-and-search student. The student copies the error message from the compiler and inserts and searches in the web browser. Both students receive an


Web Browsing & Programming Flow
Start
Tab Update (scheme pi ...)
Tab Update (Scheme 入門 ...)
Tab Update (scheme pi ...)
Tab Update (Scheme 入門 ...)
Tab Activate (Scheme 入門 ...)
Start Compiling (TaskID: 5fbe6e97e5fd1e39ee1e31bf)
Web Browsing & Programming Flow
Tab Update (scheme 変数定...)
Tab Update (scheme 変数 ...)
Tab Update (Scheme 入門 ...)
Tab Activate (Scheme 入門 ...)
Clipboard Copy (define var 1)
Start Compiling (TaskID: 5fbe6e97e5fd1e39ee1e31bf)

(A) *Student E* seek information from webpage and write code.

(B) *Student F* seek information from webpage and use clipboard copy action to submit code.

FIGURE 3.22: Group of cautious students. Students use web searches before programming. Once the solution is identified, write code from scratch or use a clipboard copy to solve the task.


Web Browsing & Programming Flow
Start
move task
move task
move task
Start Compiling (TaskID: 5fbe6e97e5fd1e39ee1e31bf)
Compile Success (3.14...)
Submit Code (TaskID: 5fbe6e97e5fd1e39ee1e31bf)
Web Browsing & Programming Flow
Start
move task
move task
move task
move task
move task
move task
move task

(A) *Student G* check few programming tasks after starting experiment.

(B) *Student H* check all programming tasks after starting experiment.

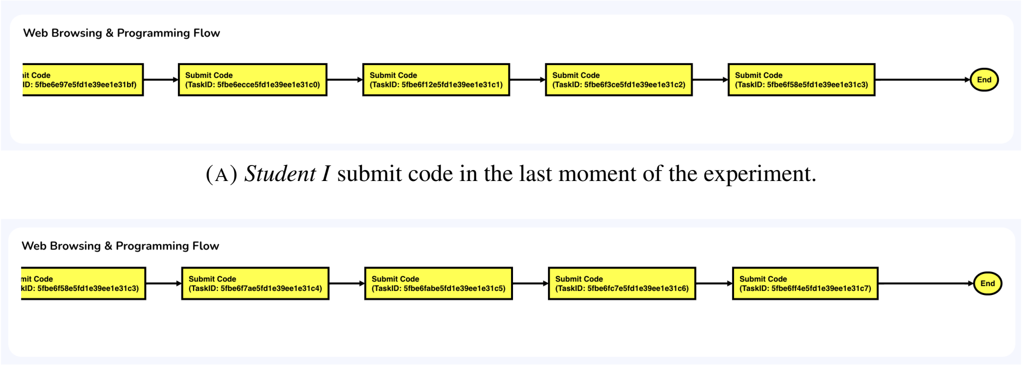
FIGURE 3.23: Group of time management students. Students move to other tasks after starting problem-solving tasks. One participant looked for a few questions, whereas the other student looked at all questions before starting to solve questions.

error response on the same problem, but each student acts differently to continue solving the problem.

Figure 3.22 shows one of the sample student group workflows for solving a programming task. A characteristic of this student is that the student first searches for information about the programming task on web pages. Once students understand the solution to the programming problem, they return to the online IDE to code and submit. Two students' workflows look similar, but one student chooses to write code from scratch, while the other uses clipboard copy. Both students are cautious about solving the problem and look for a solution before compiling. We call this group the *cautious students*.

Figure 3.23 shows an example of the student workflow for looking at multiple questions right after the experiment starts. One student looks at a few questions while others look at all the questions before starting to solve questions. This type of student did not appear among the non-lecture students and only existed among the lecture students. Lecture students have domain knowledge of the language and hence try to solve questions that are easy to solve first. We call this group *time management students*.

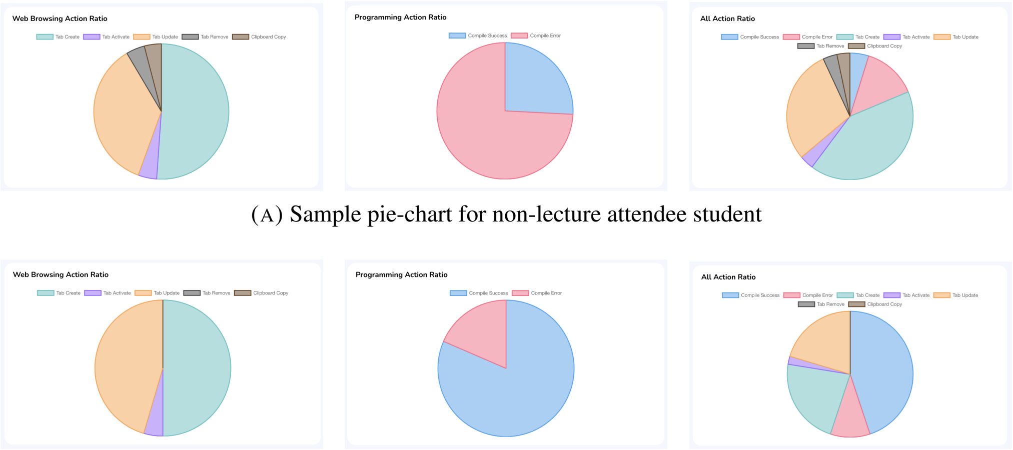
Figure 3.24 shows an example of double checking students group. Before finishing the programming problem-solving experiment, students go back to the whole programming


Web Browsing & Programming Flow
ıit Code
ID: 5fbe6e97e5fd1e39ee1e31bf)
Submit Code
(TaskID: 5fbe6ecce5fd1e39ee1e31c0)
Submit Code
(TaskID: 5fbe6f12e5fd1e39ee1e31c1)
Submit Code
(TaskID: 5fbe6f3ce5fd1e39ee1e31c2)
Submit Code
(TaskID: 5fbe6f58e5fd1e39ee1e31c3)
End
Web Browsing & Programming Flow
mit Code
kID: 5fbe6f58e5fd1e39ee1e31c3)
Submit Code
(TaskID: 5fbe6f7ae5fd1e39ee1e31c4)
Submit Code
(TaskID: 5fbe6fabe5fd1e39ee1e31c5)
Submit Code
(TaskID: 5fbe6fc7e5fd1e39ee1e31c6)
Submit Code
(TaskID: 5fbe6ff4e5fd1e39ee1e31c7)
End

(A) *Student I* submit code in the last moment of the experiment.

(B) *Student J* submit code in the last moment of the experiment.

FIGURE 3.24: Group of double checking students. Students move to previous questions to double check and submit their code before finishing the experiment.



(A) Sample pie-chart for non-lecture attendee student

(B) Sample pie-chart for lecture attendee student

FIGURE 3.25: Pie-chart for selected non-lecture and lecture attendance students. The pie-chart represents a ratio of students' action counts. The left shows an action ratio of web browsing logs, the middle shows the ratio of programming compilation result counts, and the right shows the ratio of all action counts.

question and double-check their submission. These students tend to be careful when solving questions.

Figure 3.25 shows the action ratio pie-charts obtained from the experiment. One from each non-lecture attending and lecture attending student. From the left, the pie-chart represents an action ratio for web browsing, programming, and a combination of web browsing and programming. Lecture attendee students have domain knowledge, so the success count of compilation is more than an error. Non-lecture attendee student shows the opposite. Also, the number of web searches increased for non-lecture attendance students compared to the number of lecture attendance students. This characteristic was presented earlier by previous work [174].

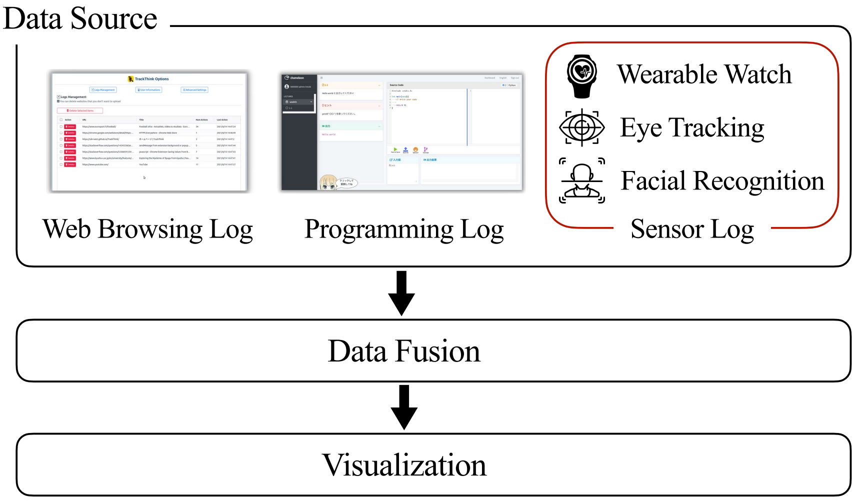


FIGURE 3.26: Future Work. Allow a variety of data sources, including plugins, to be customized as desired by teachers and students to understand students' self-regulated learning status better.

3.4.5 Discussion and Future Work

Propose a system to visualize web browsing and programming in a single chart

This study uses a flow-chart format to understand the workflow of combining web browsing and programming. This approach allows us to understand how students answer each task. Not only do we understand the web resources students use to solve tasks, but we also discover student individuality. For example, how students search when solving programming tasks. The flow-chart visualization approach supports understanding in detail how students solve the problem. Also, pie-chart visualization supports understanding actions often used in problem-solving. It helps to understand how much students understand programming language [174]. Therefore, a pie-chart supports monitoring the progress of problem-solving, and a flow-chart supports looking more carefully and specifically at how they solve questions. The work also helps to understand the students' way of thinking.

However, some future work can be highlighted for visualization. Firstly, there is the timestamp of each action. We must determine how long each action takes using the current visualization method. In this study, we sort the actions by the timestamp in order. By considering the duration of each action, for example, by changing the length of each node, the visualization could be more helpful in discovering when students are struggling with problem-solving. The flow-chart can be improved by considering feature branches such as Git [31]. We make the flow-chart into a single feature branch in the current release. We could also make two feature branches, such as web browsing and programming, to show the change in action platform from web browser to online IDE. We avoid this approach because the visualization becomes complicated and more straightforward when it is presented in a single line. Although it is possible to create a complex visualization, separating feature branches might be better in some cases. Lastly, we could add more sensor information as a data source. Figure 3.26 shows an example of sensors to be added as plugins. Wearable watches can add it as a plugin to measure heart rate for detecting stress [57]. Eye tracking can be added as a plugin to measure attention level [78]. Facial recognition can be added as a plugin to measure micro-behaviors [177] or engagement level [175]. The data fusion process allows adding any time series sensor information. Therefore, our future work will extend by adding more data sources to better understand students' cognitive states and behaviors.

Discover web browsing and programming patterns for self-regulated learning

This study discovered several patterns in students' web browsing and programming self-regulated learning. The groups we discovered are *try and error*, *try and search*, *cautious*, *time management*, and *double checking* students. Combining web browsing (knowledge input action) and programming (knowledge output action) logs, we found a unique characteristic of students' problem-solving approach. This observation is difficult to discover from the submitted code alone. The result helps teachers understand the students in-depth and gives an idea about the coaching direction. The dashboard also shows how high-scoring students solve programming problems. The solving process can be directly shared with novice students to integrate knowledge. Pie-chart supports understanding the existence of domain knowledge [174].

However, some future works can be further worked on. First, it does not take into account the combination of multiple patterns. For example, we could look at the combination of two or more patterns, such as *cautious* students might do *time management* or *double checking* pattern. By understanding the combination of patterns, we could understand more about the students' programming problem-solving process. By understanding students' multiple idiosyncratic workflow patterns, we can better understand them and help teachers support their learning. Second, it was necessary to consider the design of programming tasks that would allow for diversity in performance for each subject. In this study, the task had difficulty understanding the Scheme language's syntax, but the questions were relatively simple. Therefore, we could not receive a variety of scores from the questions. The strength of our system is that any programming language can be selected. We will discover the correlation between problem-solving patterns and scores in future work. Lastly, the segregation with GPTs [51]. In our work, we target ordinary web browse searches. Our work concerns whether web search is continuously used for the programming study. One of the future works is to allow collecting logs of prompt engineering while using chatGPT [129]. Our study focuses on more than web searching but web browsing in general. Our future work includes logging all activities, such as what kind of prompt is used in chatGPT, and analyzing how students reached their problem-solving goal in programming.

3.4.6 Conclusion

In this work, we propose TrackThink Dashboard, the system for visualizing flow charts and pie charts of self-regulated learning workflow using web browsing and programming activity logs. We collected data from 33 university students who were working on solving the Scheme language. As a result of the visualization, we discovered several unique problem-solving patterns. This research allows students and teachers to receive further feedback on the self-regulated learning process to improve study efficiency.

Chapter 4

Knowledge Transfer Activity Recognition

This chapter explains the approach of knowledge transfer activity recognition. In communication, there are certain significant activities for estimating knowledge transfer.

Section 4.1 explain *DisCaas* [177] project, aims to recognize a physical activity, especially micro-behavior. The target activities are *speaking* and *nodding*. We aim to estimate the activity only from the video recording images.

Section 4.2 explain *EnGauge* [175] project aims to recognize the cognitive state of participants. The target cognitive state is the participant engagement. We estimate the three-level engagements using the camera as a sensor with deep learning.

Section 4.3 aims to compare how gender as a feature performs in emotion recognition. We use the open-source emotion dataset to evaluate if consideration of gender makes classification result of *boredom*, *confusion*, *frustration*, and *engagement*. Emotion detection performed well after applying the gender perspective.

Section 4.4 explains the project's aim to estimate the comprehension level of participants while watching a video lecture. The study collects eye-tracking data using an eye-tracker. Ground truth is collected by asking participants to make confident annotations after watching each video. Comprehension is collected using post questionnaires.

Section 4.5 explains the project's aim of estimating presentation skills using a camera from facial and body movements. The data is collected from university lectures. Students in the lecture are asked to make a presentation. Both the lecturer and audience evaluate the presentation recorded using a camera and skill of presentation.

Overall, this chapter shows that physical activity and cognitive state recognition are significant for knowledge transfer. Targeting specifically on *nodding*, *speaking*, *engagement*, *boredom*, *confusion*, and *frustration*. Also, the studies challenge the estimation of comprehension from gaze data and presentation skills from camera recording.

4.1 DisCaaS: Micro Behavior Analysis on Discussion

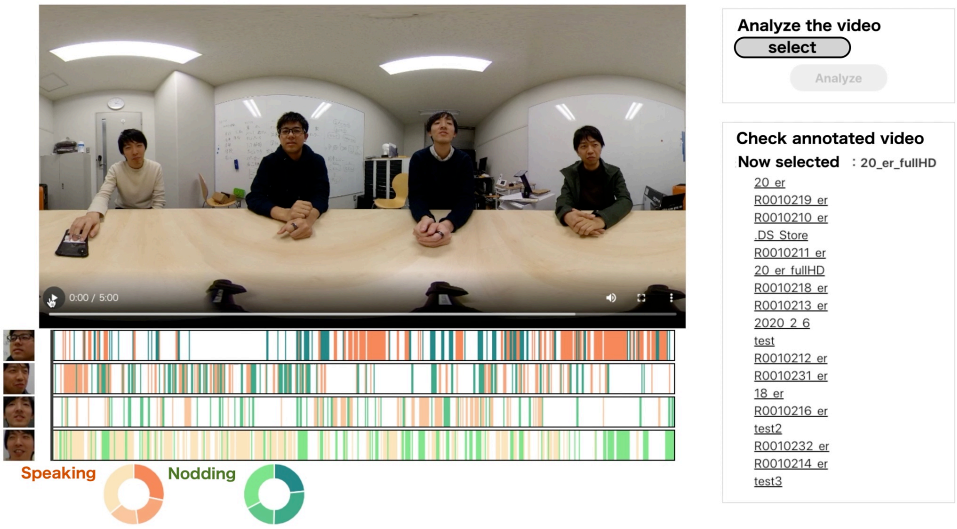
Communicating with others is one of the most important activities for generating new ideas, making rational decisions, and transferring skills. Many knowledge workers spend a certain amount of their work time on meetings. For instance, it is estimated that 11 million meetings are held in the workplace every day in the United States [7]. An employee's average time on scheduled meetings per week is six hours, and supervisors spend 23 hours [141]. The amount is increasing annually. So far, we know that between the 1960s and 1980s, this has been doubled [141]. While there is no doubt about the importance of meetings, there is also another aspect in that they are time-consuming for the participants and make up large costs for organizations, ranging from USD 30 million to over USD 100 million per year [143].

Regarding these facts, researchers have investigated how to increase the efficiency and quality of meetings [101, 102, 122, 132, 160]. After a survey of publications in social science and human-computer interaction, we found that appearances (characteristics such as age and role) [149], verbal information (e.g., spoken content/context and audio characteristics) [20, 95, 112, 113, 155, 179]. Non-verbal information (e.g., body gestures and facial expressions) change the behaviors of meeting participants [18, 30, 94, 115, 148]. Compared to several existing approaches for detecting/analyzing information mentioned above, we aim to design a system that does not utilize content-sensitive information, uses contactless devices, and is reproducible.

Content-sensitiveness: The content-sensitiveness issue mainly concerns the context relative to what was spoken in the meeting. If the meeting transcript leaks outside, the company faces high risks. We adopt an approach that extracts and stores only nonverbal data from the video stream during the meeting. Contactless: Contactless devices are also important for reducing the time consumed by device setup. Systems must also be reproductive in order to be utilized at any location. To achieve these aims, we chose to collect nonverbal information for our main data. Instead of using bodily attached devices, we chose a camera as a sensor. Reproducible: Using one device simplifies the system, increasing reproducibility. We suggested using a 360-degree camera placed at the center of the discussion table to cover all participants.

Figure 4.1 shows an example application that uses our micro-behavior detection method. Visualizing the timing of *speaking* and *nodding* of each meeting participant enables them to reflect upon how they were actively involved in the discussion. For instance, in the first half of the meeting, the fourth participant from the top was actively *speaking* and the second participant was agreeing by *nodding*. In the second half of the meeting, the fourth participant was agreeing with the first participant's opinion. Therefore, our proposed work will be used for creating a system for automatic micro-behavior annotation. It will be important for both offline and online meeting analysis.

In this section, we discuss how a camera plays an important role as a smart sensor to recognize key micro behaviors in a meeting. We propose a method that recognizes *speaking* and *nodding* from a video stream of face images and a Random Forest classifier. In order to evaluate the performance of the proposed method, we conducted data recording experiments



Now selected : 20_er_fullHD

FIGURE 4.1: A screenshot of a meeting review system that utilizes our micro-behavior recognition. A user selects a video file, and the system classifies micro-behaviors in time series.

at two physical conditions. The first recording consists of 16 sets of five minute meetings by 21 unique participants. The second recording includes seven sets of 10 min meeting data with the help of 12 unique participants. Due to the COVID-19 pandemic, most of the meetings are held online. By creating an additional dataset, we investigated whether similar features can be calculated from the webcams connected to the PCs of each participant. In summary, we present experimental results answering the following research hypotheses:

- RH1: A 360-degree camera can recognize multiple participants' micro-behavior in a small-size meeting.
- RH2: Meetings can be recorded anywhere, and the dataset can be mixed even if the collected place is different.
- RH3: Our camera as a sensor method can be utilized to evaluate offline and online meetings.

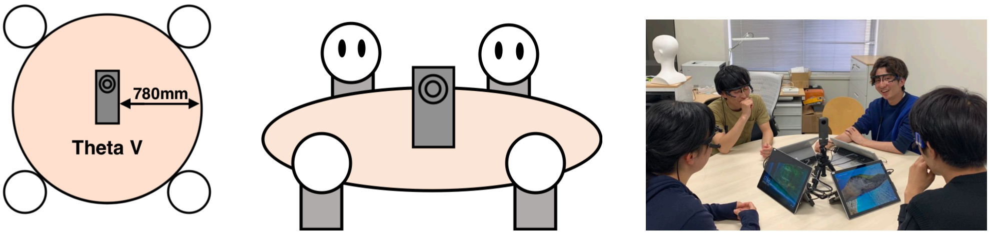
4.1.1 Architecture

This section introduces the procedures for creating a dataset and the feature extraction, detection, and classification method.

Offline Meeting Data Recording

Figure 4.2 shows an overview of our data recording setup for offline meetings. We utilized a 360-degree camera, RICOH THETA V [1]. The frame rate was 29.97 fps, and the resolution was 3840 × 1920 pixels. The camera was located at the center of a circular table. The camera records all participants in the same time series. The participant's upper body, especially the

[1] RICOH THETA V: https://theta360.com/de/about/stheta/v.html



(A) Top View (B) Side View (C) Actual Condition

FIGURE 4.2: The Device Position. RICOH THETA V is located at the center of the circular table. It is located approximately 780 mm from the edge of the table. (**A**) shows the view of the meeting condition from the top of the room. (**B**) shows the view of the meeting condition from the side of the room. (**C**) shows the actual scene of experimenting.

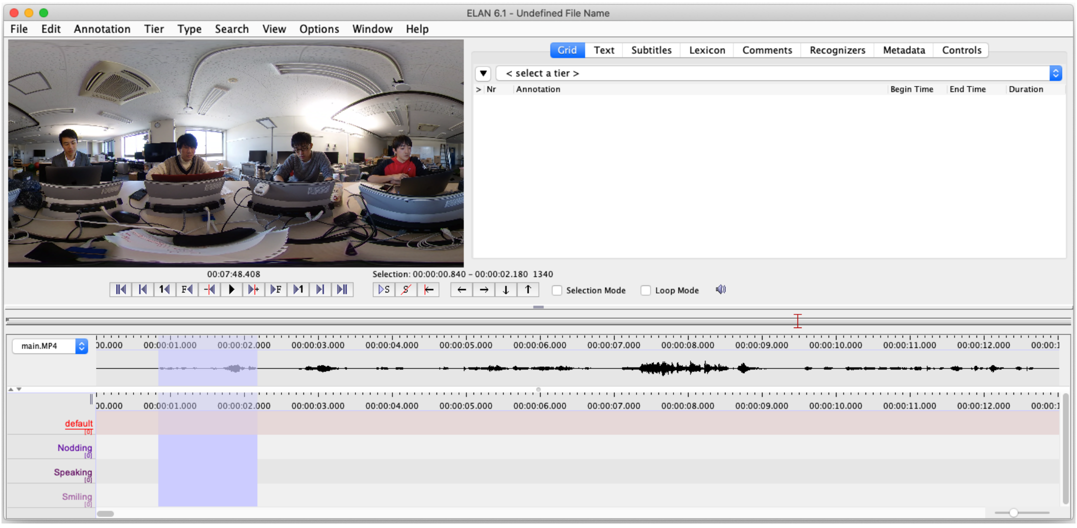

ELAN 6.1 - Undefined File Name
File Edit Annotation Tier Type Search View Options Window Help
Grid Text Subtitles Lexicon Comments Recognizers Metadata Controls
< select a tier >
Nr Annotation Begin Time End Time Duration
00:07:48.408
Selection: 00:00:00.840 - 00:00:02.180 1340
Selection Mode
Loop Mode
main.MP4
default
Nodding
Speaking
Smiling

FIGURE 4.3: Participants use ELAN to create annotations of their micro-behaviors. Each annotation contains a time duration. Annotations are performed right after each meeting session is completed. After annotation, each participant exports data into a CSV file.

face, must be seen at each trial. We decided to record the data for a maximum of ten minutes. The participants performed the annotations of each action.

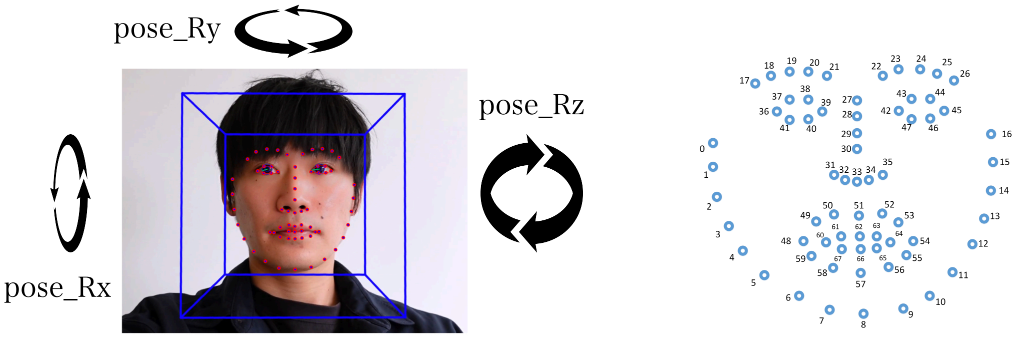
Online Meeting Data Recording

We collected data from online meetings using Google Meet [2]. The frame rate was 30.00 fps, and the resolution was 1280 × 720 pixels. A range of three to four participants joined each meeting. The meeting was held for 5 min. The participants must turn on their video each time to show their faces. Annotators perform the annotation.

Annotation of Micro-behaviors

The annotations of micro-behaviors are performed using ELAN [157] shown in Figure 4.3. ELAN is a GUI annotation tool for audio and video recordings. Users can choose to set labels for annotations. The participants were asked to annotate the duration of each micro-behavior.

[2] Google Meet: `https://meet.google.com/`



(A) OpenFace applied image

(B) OpenFace landmarks

FIGURE 4.4: Image after applying OpenFace and the 68 landmarks of the facial points.

For our approach, *nodding* and *speaking* are within the scope of the annotation. Participants annotated right after each meeting session was completed. The annotated data are extracted as a CSV file.

Extracting Head Rotations and Facial Points from Raw Video Frames Using OpenFace

We use the open-source software OpenFace [13] to obtain features of the participant's face. The person's images after applying OpenFace and the landmarks of each facial point are shown in Figure 4.4. OpenFace converts video data into several features: three head rotation data (*pose_Rx*, *pose_Ry*, and *pose_Rz*) and 68 facial points.

4.1.2 Extracting Features from the Head Rotations and Facial Points

Following our previous work, we extracted 60 features as it is listed in Table 4.1 [124]. Since we aim to extract *nodding* and *speaking*, we used particular points and rotations for each micro-behavior. A sliding window approach is used for each feature to extract each main label in the time window. Figure 4.5 visualizes the process of feature extraction. We set a window frame of 1.06 Section (32 frames) with 50% (16 frames) overlap. Each annotated label is normalized as an integer by majority voting. For example, micro-behaviors such as *nodding* and *speaking* are converted into 0 and 1. For labeling, most labeled numbers are selected as the main action occurring within the set time window.

Nodding is a human individual moving its head in the vertical direction. Hence, we focus on using the rotation feature, *pose_Rx* component shown in Figure 4.4. With the *pose_Rx*, we used the sliding window algorithm to extract features. We set a window frame of 1.06 Section (32 frames) and overlap at 50%. The detailed features extracted are shown in Table 4.1. Since we only used single features*pose_Rx*, we removed *sma*, *correlation*, and *angle*. These features are calculated using multiple features.

Speaking is the action of individuals in moving their upper and lower lips. When a person speaks, the distance between the upper and lower lips becomes larger. In order to collect this feature, we used face point numbers 62 and 66, shown in Figure 4.4. The distance between numbers 62 and 66 is the parameter. Then, we applied the sliding window algorithm. The

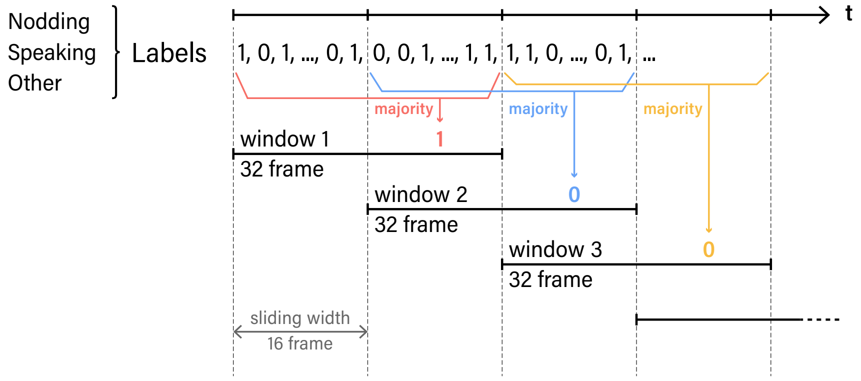


FIGURE 4.5: Sliding window algorithm used for feature extraction. In each time frame, there is a label for micro-behaviors. Label is normalized into an integer. One window is 32 frames. The sliding width is 16 frames. The label with the high majority will be a feature for each window.

window frame is 1.06 Sections (32 frames), and overlap of 50%. The features extracted are shown in Table 4.1.

Classification

For both offline and online meetings, we classified *speaking*, *nodding*, and *other* by random forest with the calculated features for each window sample. Our preliminary experiments revealed little difference in recognition performance among machine learning algorithms. Since our approach has a large number of features, we decided to use random forest, which does not degrade recognition accuracy even with a large number of features. Since comparing the performance of machine learning algorithms was not within our main scope, we only reported results using random forests in the Evaluation Section. Hyperparameters of the following are used for classification: the number of trees, 100; criterion, Gini impurity; and the number of max features, 7 (square root of the number of features).

4.1.3 Experimental Design

In order to evaluate the performance of the proposed approach, we prepared three meeting datasets. Note that our experiments do not include any EU citizens. Therefore, the General Data Protection Regulation (GDPR) does not apply to our recordings. This section explains the details of the dataset we utilized (Dataset A) and recorded (Dataset B and Online Dataset).

Offline Meeting Dataset A

Data were collected from 22 unique participants (18 males and four females) using multiple devices, including 360 cameras (RICOH THETA V). Each recording was performed for five minutes. A total of 16 sets were collected. We removed the data on participants' gaze and the acceleration data on head movements, which are included in the original dataset. The study received ethics approval (approval no: 2018-I28) after review by the Nara Institute of Science and Technology research ethics committee. This dataset is publicly available [158].

TABLE 4.1: Feature Lists.

Function	Description	Formulation	Type
mean (s)	Arithmetic mean	$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i$	T,F
std (s)	Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{s})^2}$	T,F
mad (s)	Median absolute deviation	$median_i(\lvert s_i - median_j(s_j) \rvert)$	T,F
max (s)	Largest values in array	$max_i(s_i)$	T,F
min (s)	Smallest value in array	$min_i(s_i)$	T,F
energy (s)	Average sum of the square	$\frac{1}{N}\sum_{i=1}^{N} s_i^2$	T,F
sma (s_1,s_2,s_3)	Signal magnitude area	$\frac{1}{3}\sum_{i=1}^{3}\sum_{j=1}^{N} \lvert s_{i,j} \rvert$	T,F
entropy (s)	Signal Entropy	$\sum_{i=1}^{N}(c_i \log(c_i)), c_i = s_i / \sum_{j=1}^{N} s_j$	T,F
iqr (s)	Inter quartile range	$Q3(s) - Q1(s)$	T,F
auto regression (s)	Fourth order Burg Auto regression coefficients	$a = arburg(s,4), a \in \mathbb{R}^4$	T
correlation (s_1,s_2)	Pearson Correlation coefficient	$C_{1,2} / \sqrt{C_{1,1}C_{2,2}}, C = cov(s_1, s_2)$	T
angle (s_1,s_2,s_3,v)	Angle between signal mean and vector	$\tan^{-1}(\Vert [\bar{s}_1, \bar{s}_2, \bar{s}_3] \times \upsilon \Vert, [\bar{s}_1, \bar{s}_2, \bar{s}_3] \cdot \upsilon)$	T
range (s)	Distance of the smallest and largest value	$max_i(s_i) - mix_i(s_i)$	T
rms (s)	Root square means	$\sqrt{\frac{1}{N}(s_1^2 + s_2^2 + \cdots + s_N^2)}$	T
skewness (s)	Frequency signal Skewness	$E[(\frac{s-\bar{s}}{\sigma})^3]$	F
kurtosis (s)	Frequency signal Kurtosis	$E[(s-\bar{s})^4] / E[(s-\bar{s})^2]^2$	F
maxFreqInd (s)	Largest frequency component	$argmax_i(s_i)$	F
meanFreq (s)	Frequency signal weighted average	$\sum_{i=1}^{N}(is_i) / \sum_{j=1}^{N} s_j$	F
energyBand (s,a,b)	Spectral energy of a frequency band (a, b)	$\frac{1}{a-b+1}\sum_{i=a}^{b} s_i^2$	F
psd (s)	Power spectral density	$\frac{1}{Freq}\sum_{i=1}^{N} s_i^2$	F

N: signal vector length, Q: Quartile, T: Time domain, F: Frequency domain.

Offline Meeting Dataset B

Data were collected from 12 unique participants (11 males and one female). Each recording was performed for 10 min. A total of seven sets were collected. The dataset combines 34 unique participants (29 males and five females). The total time collected is 150 (80 + 70) minutes.

Online Meeting Dataset

For online meeting analysis, we collected data using Google Meet. Our main proposal is the offline meeting analysis, but we also performed an online meeting analysis for future discussions. The data are collected from Kyushu University, Japan. Unique participants

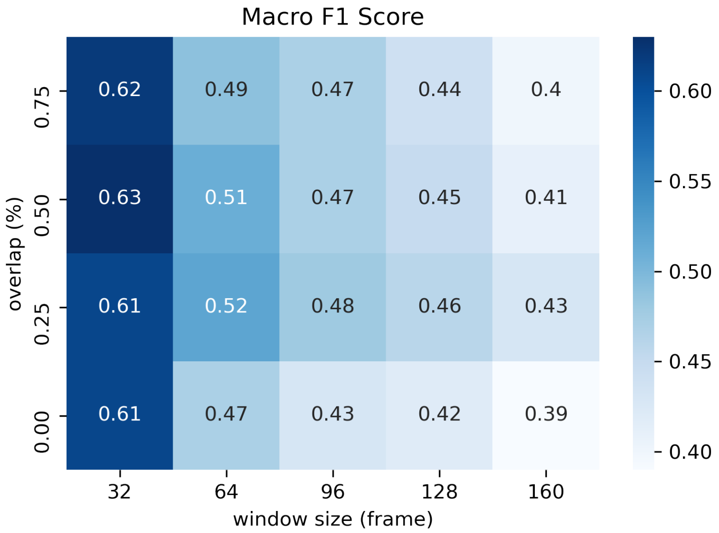


FIGURE 4.6: Heatmap of macro average F1-score of overlap vs. window size. All offline meeting datasets are used. The unit for window size is by frame. The unit for overlap is by percentage.

were six people in total, and each meeting was collected for five minutes. Seventeen sessions were collected. The total time collected amounts to 85 minutes.

Evaluation Protocol

Using the model, the 10-fold random cross-validation and leave-one-participant-out cross-validation were applied. The 10-fold random cross-validation used a one-fold random dataset as test data and the other nine-fold random dataset as training data. We used ten sets to perform cross-validation. In the case of leave-one-participant-out cross-validation, test data includes one participant, and train data are used for all others. Since the amount of data relative to the labeled behaviors (*nodding* and *speaking*) is lesser than non-labeled behavior (*other*), we used down-sampling methods. Each data is reduced to balance the actions of *nodding* or *speaking*. We used random forest for all three patterns for the machine learning technique. On the other hand, we also ran a prediction of micro-behavior analysis for online meetings. For online meetings, we classified *nodding*, *speaking*, and *other*. The 10-fold random split cross-validation and leave-one-participant-out cross-validation were applied.

4.1.4 Results and Discussion

For offline meeting analysis, we collected data from two different places. We classified each dataset and combined them, calling them “Dataset A”, “Dataset B”, and “Dataset A + B”. As it is shown in Figure 4.6, we have decided to use the window size of 1.06sec (32 frames) with 50% (16 frames) overlap because the macro average F1-score was the highest. The precision, recall and f1-score results are shown in Table 4.2. The confusion matrix of *nodding*, *speaking*, and *other* is shown in Figure 4.7. As a result, *nodding* becomes a lower f1-score than *speaking*. This result shows that *nodding* is challenging to predict compared to *speaking*. *Speaking* takes an average time of 4.01 s, and *nodding* takes an average of 1.06 s. The

TABLE 4.2: Prediction result of *Nodding* and *Speaking* for Offline Meeting.

	(a) 10-Fold Random Split			
Dataset	Label	Precision	Recall	F1-Score
A	*nodding*	0.66 ± 0.09	0.58 ± 0.17	0.61 ± 0.13
	speaking	0.68 ± 0.07	0.75 ± 0.07	0.71 ± 0.05
	macro ave.			0.65 ± 0.05
B	*nodding*	0.73 ± 0.04	0.62 ± 0.15	0.66 ± 0.10
	speaking	0.69 ± 0.04	0.78 ± 0.03	0.73 ± 0.03
	macro ave.			0.69 ± 0.05
A + B	*nodding*	0.68 ± 0.10	0.61 ± 0.17	0.64 ± 0.14
	speaking	0.69 ± 0.05	0.75 ± 0.04	0.72 ± 0.04
	macro ave.			0.68 ± 0.07

	(b) Leave-One-Participant-Out			
Dataset	Label	Precision	Recall	F1-Score
A	*nodding*	0.64 ± 0.17	0.60 ± 0.17	0.60 ± 0.14
	speaking	0.63 ± 0.21	0.68 ± 0.21	0.63 ± 0.19
	macro ave.			0.62 ± 0.09
B	*nodding*	0.60 ± 0.25	0.49 ± 0.25	0.53 ± 0.23
	speaking	0.57 ± 0.26	0.64 ± 0.24	0.58 ± 0.24
	macro ave.			0.58 ± 0.15
A + B	*nodding*	0.66 ± 0.16	0.59 ± 0.19	0.60 ± 0.17
	speaking	0.65 ± 0.20	0.71 ± 0.19	0.66 ± 0.18
	macro ave.			0.63 ± 0.11

TABLE 4.3: Prediction Result of *Speaking* and *Nodding* for Online Meeting.

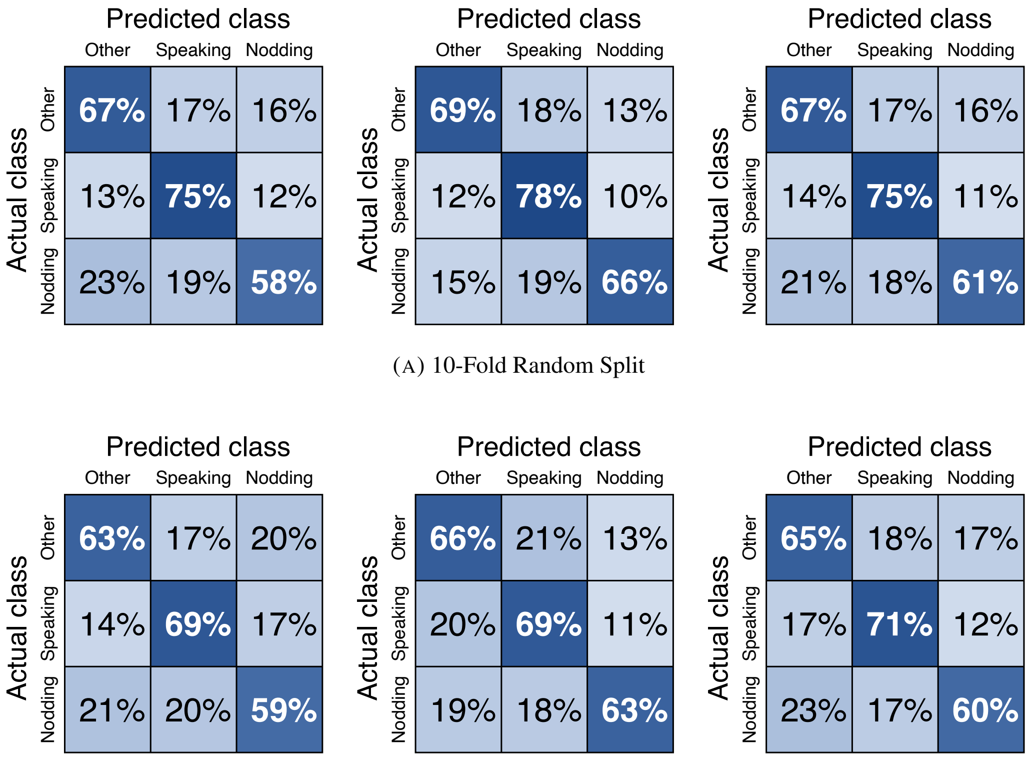
(a) 10-Fold Random Split

Label	Precision	Recall	F1-Score
nodding	0.66 ± 0.17	0.60 ± 0.17	0.60 ± 0.14
speaking	0.62 ± 0.22	0.68 ± 0.19	0.64 ± 0.19
macro ave.			0.55 ± 0.08

(b) Leave-One-Participant-Out

Label	Precision	Recall	F1-Score
nodding	0.41 ± 0.26	0.23 ± 0.13	0.23 ± 0.14
speaking	0.37 ± 0.29	0.40 ± 0.25	0.31 ± 0.10
macro ave.			0.31 ± 0.01

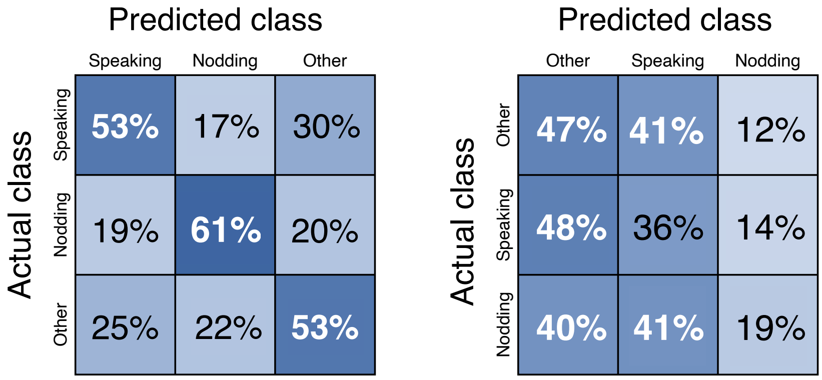
results of macro average f1-score for 10-fold random split cross validation are 0.69 ± 0.05, and 0.68 ± 0.07. Dataset B is the highest among the three dataset patterns. The results of the macro average f1-score for leave-one-participant-out cross validation are0.62 ± 0.09,



(A) 10-Fold Random Split

(B) Leave-One-Participant-Out

FIGURE 4.7: Confusion Matrix of *nodding*, *speaking*, and *other*. Using facial points and head rotation data as features. Downsampling was applied. Only offline meeting datasets were used. From the left figure, the result is extracted from M3B Corpus [158] (Dataset A), Kyushu University (Dataset B), and both (Dataset A + B). Datasets are split into two patterns: (**a**) 10-fold random split and (**b**) leave-one-participant-out. Random forest is used as the machine learning algorithm.



(A) 10-Fold Random Split

(B) Leave-One-Participant-Out

FIGURE 4.8: Confusion matrix of *speaking*, *nodding*, and *other*. Facial points and head rotation data were used as features. Downsampling was applied. Only online meeting datasets were used. The dataset was split into two patterns: (**a**) 10-fold random split and (**b**) Leave-One-Participant-Out. Random forest was used as the machine learning algorithm.

0.58 ± 0.15, and 0.63 ± 0.11. Dataset A + B was the highest among the three dataset patterns. In the case of leave-one-participant-out cross-validation, the lowest macro average f1 score is 0.39, and the highest is 0.78.

TABLE 4.4: Feature Importance of Micro-Behavior Recognition.

(a) Offline Meeting

Rank	Function	Component	Type	Weight
1	iqr	distance between facial point 62 and 66	frequency	0.046
2	iqr	pose_Rx	frequency	0.040
3	std	distance between facial point 62 and 66	time	0.039
4	ARCoeff-2	distance between facial point 62 and 66	time	0.038
5	ARCoeff-1	pose_Rx	time	0.037

(b) Online Meeting

Rank	Function	Component	Type	Weight
1	entropy	pose_Rx	time	0.033
2	mean	pose_Rx	time	0.031
3	ARCoeff-3	distance between facial point 62 and 66	time	0.030
4	min	pose_Rx	time	0.029
5	Skewness-1	pose_Rx	frequency	0.027

For the online meeting, the precision, recall, and f1-score results are shown in Table 4.3. The confusion matrix of *nodding*, *speaking*, and *other* is shown in Figure 4.8. The result of the macro average f1-score of the 10-fold random split cross-validation is $\mathbf{0.55 \pm 0.08}$. The result for the leave-one-participant-out cross-validation is $\mathbf{0.31 \pm 0.01}$. In the case of the leave-one-participant-out cross-validation, the lowest macro average f1-score is **0.27** and the highest is **0.35**. The calculation of each function is explained in Table 4.1.

Regarding Table 4.4, the results show that, for offline and online meetings, the feature importance was different among them. The feature importance of the offline meeting is shown in Table 4.4 a, and the online meeting is shown in Table 4.4b. We have found that the most important features of offline meetings are the components related to lips. For online meetings, the most important features are the components related to head rotation.

4.1.5 Discussion

In this section, we discuss the three research hypotheses stated.

Can a 360-degree camera recognize multiple participants' micro-behavior in a meeting?

Combining all datasets, the macro average f1-score is $\mathbf{0.68 \pm 0.07}$ for the 10-fold random split cross-validation. The leave-one-participant-out approach macro average f1-score is $\mathbf{0.63 \pm 0.11}$. We found out that we could collect micro-behaviors in the meeting by using only a 360-degree camera as a sensor. Multinational classification is possible. The f1-score of *speaking* is the highest in any condition. *Speaking* recorded a higher score because each participant's action does not vary. All participants opened their mouths while *speaking*. Therefore, using the distance of the upper and lower lip as the feature is practical. On the other hand, *nodding* seems different between each participant. When we looked at the raw video data, we discovered that some participants perform *nodding* with shallow and fast head movements, while others only perform deep and fast nodding. Concerning this, using head rotation data for predicting *nodding* scored lower than *speaking*. We also discovered that for the leave-one-participant-out approach, participants' lowest macro average f1-score was **0.39** and the highest is **0.78**. When we looked at the annotation data, we discovered that participants with the lowest f1-score had fewer labels of *nodding* than the highest participant. This is due to the difference in annotations. The participants with fewer annotations were only labeled with *nodding* that was deep and slow. This result caused less feature data for *nodding*, reducing the score. Overall, the discussion states that *speaking* recognition from nonverbal data is more accessible than *nodding*. In terms of the answer for *RH1*, the macro average of the f1-score achieved $\mathbf{0.68 \pm 0.07}$ and $\mathbf{0.63 \pm 0.11}$; we could say that using 360 camera as a sensor for detecting micro-behaviors is effective.

Can we extend the dataset by adding data recorded in other places?

In order to prove the possibility of the expanding dataset, we collected data from two different locations. The results of 10-fold random split cross-validation showed that each dataset produced a f1-score of $\mathbf{0.65 \pm 0.05}$, $\mathbf{0.69 \pm 0.05}$, and $\mathbf{0.68 \pm 0.07}$, shown in Table 4.2. The result was the highest for Dataset B. Our assumption regarding this result is the number of the same participants included in the dataset. In Dataset A, 22 unique participants joined. Among them, 12 participants were involved in 20 minutes of the meeting, and ten participants were involved in ten minutes in total. For Dataset B, 12 unique participants joined. Among them, one participant joined for 50 minutes, two participants joined for 40 minutes, two participants joined for 30 minutes, two participants joined for 20 minutes, and three participants joined for ten minutes. By comparing Datasets A and B, the duration of each participant's participation in the meetings is longer in Dataset B. This means that the volume of individual behavior in Dataset B is the largest. Hence, the model accuracy is the highest for Dataset B without removing participant behavior information as test data. Adding more personal data will improve the prediction rate for each person when testing. With this result in mind, the leave-one-participant-out cross validation produced the f1-scores for each dataset of the following: $\mathbf{0.62 \pm 0.09}$, $\mathbf{0.58 \pm 0.15}$, and $\mathbf{0.63 \pm 0.11}$. We have found that Dataset B is the lowest in the f1-score compared to the analysis of another dataset. Once personal data are removed from the dataset, the f1-score slightly decreases. However, it is interesting that combining all datasets' f1-score results in $\mathbf{0.63 \pm 0.11}$. This result is the highest compared to the prediction of the other dataset. For the answer for *RH2*, the results show that the more the dataset increases, the accuracy of the model of micro-behavior prediction also increases.

Can our camera as a sensor method cover both offline and online meetings?

The result from Table 4.2 shows that the 10-fold random split cross-validation produced high f1-scores. The result of f1-score is $\mathbf{0.55 \pm 0.08}$. For the leave-one-participant-out cross validation, the score is $\mathbf{0.31 \pm 0.01}$. The classification result decreases with the leave-one-participant-out cross-validation. This is probably the result of the same reason stated in the discussion of *RH1*. Individual data will be powerful for predicting certain people's behavior. By looking at the result of the f1-score for the individual participants, it is observed that the lowest is **0.27** and the highest is **0.35**. One of the unique aspects of the reduction in F1 scores in online meetings is that it is caused by the position of the face. Our proposed method for offline meeting analysis uses a 360-degree camera to track the entire upper body of the participants, but for online meetings, the participant's face is often the only object recorded. Regarding fewer white spaces for face tracking, online meetings often became off track when using OpenFace. For the answer relative to *RH3*, even with the concern of failure in tracking the face, the f1-score of $\mathbf{0.55 \pm 0.08}$ and $\mathbf{0.31 \pm 0.01}$ says that our camera can be potentially used as a sensor approach for online meetings.

4.1.6 Limitations and Future work

The imitations of our work include recognizing multiple actions at the same time. Our models only predict a single behavior happening in a set time range. However, the meeting behavior is complex. *Speaking* and *nodding* can happen simultaneously, but we did not consider predicting both simultaneously.

Another consideration is the nervous tic of *nodding*. Some participants perform more *nodding* actions than others. However, we cannot detect whether the *nodding* performed by participants includes some context. We only detected the movement of *nodding*. Therefore, our future work includes recognition of the context inside *nodding* performed by participants.

We also have to mention the meeting's time, duration, and size. We conducted meetings of ten minutes, with four people in each meeting as the largest size. In theory, our approach can be used even after long meetings. However, we only recorded meetings for a maximum of 10 minutes since we considered that the load relative to annotations for each meeting participant would be high. Our approach uses a sliding window, which means we split the video into a set length so that the total duration of meetings will not be a problem. In theory, our approach can be used even in meetings of a larger size if we can capture all participants' faces. For offline meetings, we have limited physical space. If we want to increase the number of participants, we must add more cameras. For online meetings, the maximum number of participants displayed on a screen is limited to the device screen. Moreover, as the number of participants increases, the screen size of each participant will be smaller. Hence, for future work, we need to think about how to record the meetings with more participants.

One limitation of online video compared with offline and online meetings is the angle of view. Using a 360-degree camera, we consistently tracked the participant's upper body. We can usually track the participants' faces during the experiment. However, for the online meeting, the angle of view differs for each participant. In particular, when participants try to observe the screen shared on their laptops, they often move closer to the screen. In those cases, the participant's face moves outside the box, making it impossible to track their face. Moreover, notifications often distract participants' faces. It covers the participants' faces, which results in being unable to track the participant once someone types into the chat.

Since we extracted faces from screen recordings, the recording condition was not constant. Our face tracking failed when the face image was hidden by a desktop notification or became relatively smaller when someone started sharing a screen. This problem should be solved in our future work. For instance, making our application, bot service, or plugin for an online meeting tool are all potential directions.

For the machine learning model, we must consider the importance of features. We have found that the importance of the machine learning model's features varies offline and online. Hence, the method is the same, but we need to collect the dataset of each offline and online meeting in order to create a precise prediction model of *speaking* and *nodding*. Moreover, processing time is not considered in our work, and extending our work into real-time micro-behavior recognition will be important. Regarding these findings, the experiment setting for online meeting analysis could be further explored for future work.

4.1.7 Conclusion

In this work, we analyzed micro-behaviors during offline and online meetings. For offline meeting analysis, we used 360-degree cameras, and for online meeting analysis, we used Google Meet. Our target micro-behaviors are *speaking* and *nodding*. For the offline meetings, the result of the f1-score is **67.9%** for 10-fold random split cross-validation. For the leave-one-participant-out cross-validation, it is **62.5%**. We also discovered that combining the dataset collected in different places can still increase the accuracy of the recognition model for offline meeting data. This result suggests that anyone can follow our work as a framework to increase the dataset and accuracy of the model. We also applied cameras as a sensor method for the online meeting. For the 10-fold random split cross-validation, we observed a macro average f1-score of **55.3%**. The leave-one-participant-out approach achieved a macro average f1-score of **31.1%**. We discovered that participant behavior is important for creating an accurate model. Micro-behavior analysis using a camera as a sensor approach will become the means for new meeting analysis platforms.

4.2 EnGauge: Engagement Gauge of Meeting Participants

Engagement is an important factor in deepening a person's level of learning [52, 140, 150]. Due to the importance of engagement in learning effectiveness, several studies have examined ways to estimate student engagement during lectures [45]. Previous studies have used multiple sensors such as wristband [45] or seat sensor [56] to discover participants' engagement levels during offline lectures. However, due to the COVID-19 pandemic, offline meetings such as lectures, office meetings, job interviews, and musical events have expanded online [126]. Although participants see each other in an online meeting, they receive less information than in offline meetings [165]. For example, there is a limitation in computer screen size. Thus, only the upper body of a limited number of participants could be observed. Hence, the received nonverbal information, such as hand and body movements and gestures, is reduced. Furthermore, technical inputs such as a microphone, speaker, and computer screen are involved in verbal information. Limitations of these technical inputs sometimes lead to interruptions. For example, a faint voice may result in difficulties emphasizing speech. In educational settings, such as e-learning, where many participants are involved in contrast to a few speakers, it is difficult for speakers or teachers to keep track of the participants' situations and technical and other difficulties. In such cases, an information gap between the speakers and listeners may encourage them to participate in other tasks.

Our work is inspired by the *EduSense* project established by Ahuja *et al.* [6]. This study focuses on using a camera as a sensor to collect students' activities in offline classes. The system uses Microsoft Kinect's one-depth camera and Intel NUC to discover student activities such as raising a hand, sitting, standing, smiling, speaking, attention, class gaze, or head orientation. Then, we further step into the engagement by referring to the work done by Dhamija. They collected data using a Logitech webcam with a 640×480 resolution at 30 frame-per-second and contained five-scale self-reported engagement levels as truth labels [44]. They proposed a facial information method to classify participants as engaged or disengaged. The participants were asked to watch a neutral introductory video in their study. Like their work, De Carolis *et al.* conducted a video meeting engagement analysis while watching e-learning content [42]. These previous studies inspired our work to focus not only on students watching the online video content but also on analyzing online meetings. Therefore, we aimed to classify participants' engagement levels in online meetings. In previous works, annotation of engagement levels was done either by self-reporting or labeling annotators. Both works are time-consuming and cannot accurately label the engagement level, especially for the annotator labeling. Therefore, in our study, we collected label data by asking the participants to role-act at each engagement level.

This study focused on participants' engagement levels in e-learning and educational settings. Students join online classes or meetings using desktop computers or laptops with webcam microphones. During these online sessions, students listen to the teacher or speaker and take notes on notebooks or devices. Our study aimed to quantify participants' meeting engagement levels. In particular, we focused on three points to make our quantification system more realistic. First, we only used a built-in camera on a personal computer instead of an advanced sensor. Second, we focused on identifying engagement levels similar to those of real-life online meetings, where the participants perform natural actions such as speaking, taking notes, or reading. Third, we only used nonverbal information to safeguard privacy and avoid recording the context of information spoken during meetings. While considering three points, our contributions to this paper are the following:

1. **Uniqueness of the data collection**: Approach to collecting the variant engagement levels. We select a role-acting method to collect three engagement levels in this work. Related work selects post annotation, such as self or non-self human annotation or questionnaires.

2. **Best performed engagement level detection model**: Using a webcam as a sensor, we classify high, middle, and low engagement levels with the F1 score of 0.895. The accuracy was the best compared with previous work. We use the model of *MobileNetV2*, and the input data is only a facial image.

3. **Implementation of the application and conducting a pilot study**: We implement the engagement detection system called *EnGauge* using our own implemented deep learning model. The application shows the engagement level as 0 to 100 with a gauge-like interface. We also conduct a pilot study with the system.

4.2.1 Architecture

The ultimate goal of our study is to estimate participant engagement during online meetings with high accuracy from camera information only. For this purpose, we utilize several deep learning-based models in addition to the conventional sliding window method and compare their accuracy. Finally, we will build an application using the model with the highest recognition accuracy.

Feature extraction-based engagement estimation

We processed these raw video data using the OpenFace tool to extract features, including three head rotation data (pose_Rx, pose_Ry, and pose_Rz) and 68 facial landmark coordinates. Sometimes, the participant's face was out of the screen, and OpenFace could not detect facial features. We excluded undefined data from our experiment in such cases. OpenFace generated an output CSV containing facial features at the end of preprocessing. We also annotated the frames in the final dataset with the corresponding ID and high, middle, and low engagement levels.

Nakamura *et al.* identified 60 features divided into the time and frequency domains from different time-series input data captured by sensors [124]. In the previous research regarding micro-behavior analysis using a computer camera as an input sensor, Watanabe *et al.* also utilized these features for vectorizing facial feature coordinates and rotations input data to extract nodding and speaking behavior [177]. Similarly, we also used head rotation and facial landmark data for further processing. From the head rotation (*pose_Rx*, *pose_Ry*, *pose_Rz*) and 68 facial landmark locations output of the OpenFace framework, we converted these

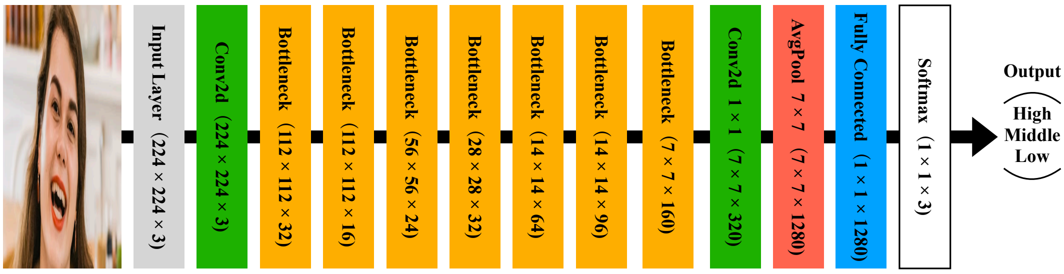

Input Layer (224 × 224 × 3)
Conv2d (224 × 224 × 3)
Bottleneck (112 × 112 × 32)
Bottleneck (112 × 112 × 16)
Bottleneck (56 × 56 × 24)
Bottleneck (28 × 28 × 32)
Bottleneck (14 × 14 × 64)
Bottleneck (14 × 14 × 96)
Bottleneck (7 × 7 × 160)
Conv2d 1 × 1 (7 × 7 × 320)
AvgPool 7 × 7 (7 × 7 × 1280)
Fully Connected (1 × 1 × 1280)
Softmax (1 × 1 × 3)
Output
High
Middle
Low

FIGURE 4.9: Architecture of MobileNetV2.

points into feature vectors of time and frequency domain features listed in the mentioned table. For the vector feature extraction process, we applied a sliding window algorithm on a randomly chosen specific number of frames (i.e., 32, 64, 128, 256, 500, and 1000) with 50% overlap to calculate the maximum frequency of annotated engagement labels by taking advantage of majority voting. We decided to drop the *NaN* values instead of filling them with any integer or float values because that would impact the final result. So, nearly 20% *NaN* values were dropped.

For the machine learning model, we applied Gaussian Naive Bayes, Decision Tree, and Random Forest to the combined feature vectors. The model used the following hyperparameters: the number of trees, 100; criterion, Gini impurity; and maximum features, 7 (square root of the number of features).

Deep learning-based engagement estimation

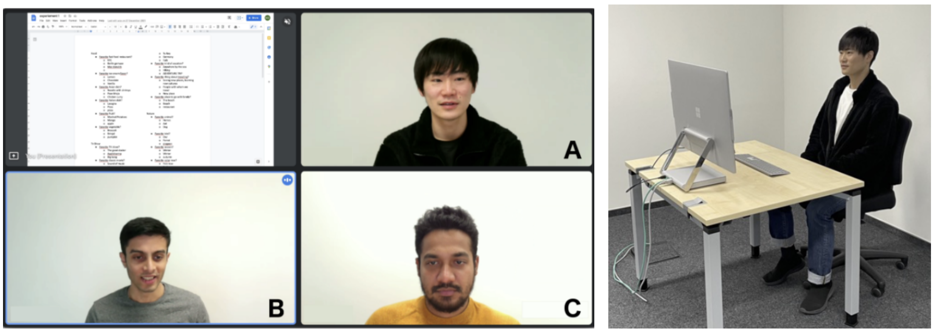
To preprocess the data, we utilized OpenCV, an open-source software library for performing machine vision tasks, to convert videos to images [19]. Specifically, we used a tool to convert videos to individual frame images with a frame rate of one frame per second. Images lacking clear faces were removed from the dataset. We converted the remaining images to grayscale and applied image hashing to remove duplicates. We trained and evaluated four models for our classification task: VGG16, Xception, MobileNetV1, and MobileNetV2.

VGG16 [156] – In our study, we utilized the pre-trained VGG16 model [156] on the ImageNet dataset. For fine-tuning, a density of three and softmax activation for the classifier were employed. Additionally, the top and two layers above the model were fine-tuned.

Xception [37] – The model was trained on the ImageNet dataset. Next, global average pooling was incorporated, and a dense layer consisting of 512 units with ReLU activation was added. The classifier was activated using softmax, and the model's top layer and blocks 13 and 14 were fine-tuned.

MobileNetV1 [73] – The model was pre-trained on the ImageNet dataset, with pre-trained layers included for fine-tuning the size was reshaped to 1024. A single dense layer was utilized, and its weight was transferred to the final six layers. The activation function used was softmax, and the last 23 layers were trainable.

MobileNetV2 [146] – The model, an updated version of MobileNetV1, is depicted in detail in Fig. 4.9. We pre-trained our implementation using the ImageNet dataset. The input layer modifies the image before passing it through two convolutional and fully connected



(A) Screenshot of the Google Meet video recording.

(B) Webcam recording experiment environment.

FIGURE 4.10: Experimental setting with participants A, B, and C assigned to high, middle, and low engagement roles, respectively.User is in front of the computer with a webcam.

layers. Finally, a softmax layer with three neurons representing high, middle, and low classes was appended to the model.

Model accuracy comparison

We evaluated the performance of classifiers using a leave-one recording out cross-validation approach. Specifically, one recording involving three participants was designated as the test data, while the remaining seven were used for training the model. This process was repeated until each recording was utilized for testing. Furthermore, in our deep learning-based approach, we randomly split the seven datasets into five and two subsets for training and validation. This evaluation method was conducted independently, as all participants attended a single meeting without overlap.

4.2.2 Data collection and ground truth labeling

We used a built-in camera to record the video of each participant while conducting online meetings. We used the online group meeting platform Google Meet for the experiment. An example of the experimental setup is shown in Fig. 4.10. The frame rate was 30.00 frame-per-second, and the screen resolution was 1280×720 pixels. They removed the face mask and turned on the camera before the discussion. We captured the screen recording of each session but excluded voice information owing to privacy issues.

Participants

In this study, we recruit 24 participants (17 male and seven female). Nationalities are South Asia, Southeast Asia, Eastern Europe, North Africa, and North America. Participants were between 22 and 36 years, and the average was 26. They were either university students or workers in Germany. Before the experiment, we obtained consent from the participants regarding the General Data Protection Regulation (GDPR). The participants were allowed to opt out of the experiment at any time. At the end of the experiment, all participants who completed it received a ten-euro Amazon voucher.

Data Collection Procedure

In this study, we set up an experiment in which participants engage in online meetings and discussion-like sessions. The experiment was conducted in a secure and controlled manner. We designed an experiment with three participants per session to create a balanced data set for engagement analysis. Each session lasted five minutes. In the experiment, each of the three participants was instructed to alternate between different engagement levels of behavior. The high, middle, and low engagement requirements are the following:

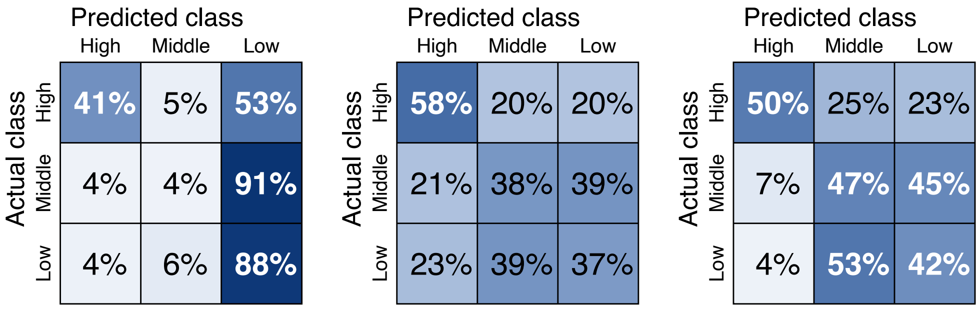
High Engagement - Subjects are prohibited from doing distracting tasks and instructed to focus only on the discussion. The main task is to answer the question asked by the middle engagement level role participant. Example behaviors observed are *answering question*, *listening to the questions*, *smiling*, *touching face*, *laughing*, *nodding*, *shaking head*, and *nodding and shaking heads*.

Middle Engagement - Subjects are prohibited from doing distracting tasks and instructed to moderate the discussion by asking questions to the high engagement level role participant and taking note of the answer in the text file document presented on the computer screen. Example behaviors observed are *asking questions*, *looking at a keyboard*, *typing on a keyboard*, *listen to the answer*, *reading notes*, *nodding*, *shaking head*, *nodding and shaking heads*, *touching face*, and *laughing*.

Low Engagement - Subjects work on a distraction task while listening to the conversations between high and middle-engagement level participants. The distraction task is to read a scientific article and write a summary of it. The paper and the notes are on the same computer as the online video meeting. Example behaviors observed are *reading a paper*, *write a summary of a paper*, *ignore the conversation*, *looking at a keyboard*, *typing on a keyboard*, *touching face*, *stretching*, *head on hands*, *playing with hairs*, *yawing*, and *clean nose*.

The specific example behaviors for each engagement role are indicated, such as *asking questions*, *answering questions*, or *reading a paper*. Rather than using self-annotation [117], observer (additional annotator) annotation [117], survey [45], or using real-time notifications to get feedback from participant [10] to determine the level of engagement of participants, we came up with the idea of asking them to work in the role acting of each engagement level. This idea was adopted because of the following concerns. The questionnaire method could blur the definition of engagement levels due to individual differences in labeling, and the method of asking participants to provide real-time feedback on their engagement levels during the experiment would have distracted most participants from providing feedback and reduced their ability to concentrate, making uniform that data collection might not be possible. Therefore, the method used in this study was to have the participants participate in the meeting in the same role from the beginning to the end of the experiment. They repeated until two sets of the high, middle, and low engagement level datasets were obtained from each participant. Hence, the experimental procedure consisted of six sessions discussing specific topics.

We prepared two topics for the meetings. The first one contained questions about the favors, such as "What is your favorite X," where words, such as fast-food restaurant, movie,



	High	Middle	Low
High	41%	5%	53%
Middle	4%	4%	91%
Low	4%	6%	88%

(A) Gaussian Naive Bayes

	High	Middle	Low
High	58%	20%	20%
Middle	21%	38%	39%
Low	23%	39%	37%

(B) Decision Tree

	High	Middle	Low
High	50%	25%	23%
Middle	7%	47%	45%
Low	4%	53%	42%

(C) Random Forest

FIGURE 4.11: Confusion matrix of feature extraction based engagement estimations with 1000-frame sliding windows.

TABLE 4.5: Comparison between F1 score and window size (frame) of the result leave one recording out cross validation for each model Gaussian Naive Bayes (GNB), Decision Tree (DT), and Random Forest (RF).

Window Size	32	64	128	256	500	1000
GNB	0.325	0.310	0.315	0.339	0.365	0.384
DT	0.350	0.354	0.356	0.388	0.419	0.446
RF	0.347	0.344	0.357	0.400	0.439	**0.467**

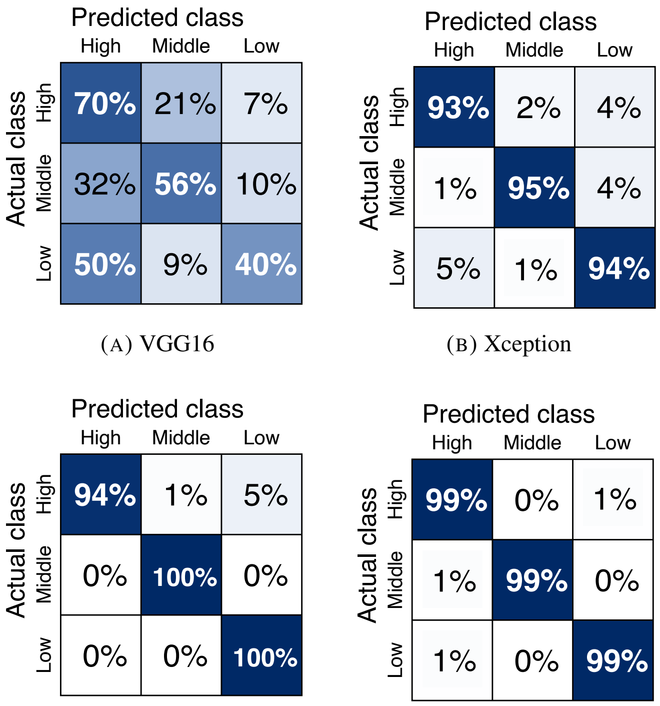
and sports, replaced 'X.' The second one contained questions about participants' current studies, such as "What do you think about your work or study?". Each group of participants was assigned an engagement role in a cyclic order. Thus, every participant was asked twice to perform the high, middle, and low engagement-level tasks. Hence, there was a total dataset of six sessions (30-minute video) for each participant. It took roughly half a minute after the first recording session for participants to understand the assigned tasks clearly. Additionally, participants were joyful during the discussion in Phase 1. Furthermore, we observed that a few participants were unhappy during the summary task because they needed more time to finish summarizing the paper within five minutes.

Results

In this section, we present the accuracy of the three engagement-level classifications. First, we explain the classification rate of the feature extraction-based approach in Section 4.2.2. Then, we explain the deep learning-based classification approach in Section 4.2.2. After that, we compare the feature extraction-based and deep learning-based models' accuracy in Section 4.2.2. Lastly, we present the result of leave-one-out-cross-validation for the highest accuracy model MobileNetV2 in Section 4.2.2.

Feature extraction based engagement estimation

We show the results of the feature extraction-based engagement estimation. Table 4.5 presents detailed results for each window size. The lowest classification rate was 32 frames (1.0 s)



(A) VGG16

(B) Xception

(C) MobileNetV1

(D) MobileNetV2

FIGURE 4.12: Confusion matrix of deep learning based engagement estimations.

TABLE 4.6: Comparisons of F1 scores in leave-one-group-out cross-validation (LOGOCV).

Test dataset (group)	1	2	3	4	5	6	7	8	Mean
Random Forest	0.264	0.464	0.521	0.476	0.520	0.519	0.429	0.546	0.467
VGG 16	0.347	0.562	0.623	0.545	0.612	0.602	0.512	0.589	0.549
Xception	0.895	0.975	0.902	0.748	0.957	0.961	0.821	0.981	0.905
MobileNetV1	0.898	0.981	0.911	0.755	0.982	0.980	0.833	0.996	0.917
MobileNetV2	0.901	0.999	0.915	0.761	0.985	0.985	0.835	0.999	**0.923**

with a 50% overlap for all types of models. By comparing several window frame sizes, we discovered that a window size of 1000 frames (34.6 s) with a 50% overlap yielded the best classification result. As the window size increased, the classification rate increased. Compared with these three models, the highest F1 score was recorded by Random Forest with 0.467 in our case study. To further verify this result, a confusion matrix was created with a window size of 1000 frames, Fig. 4.11. We could verify that middle and low-engagement level classifications often need clarification according to the confusion matrix of Random Forest. We will discuss later why middle and low-engagement-level classifications need clarification.

Deep learning based engagement estimation

We present the results of the deep learning-based engagement estimation. When the experiment was conducted, it was found that the participants sometimes moved away from the camera screen, making it impossible to track their faces. Therefore, all these images were

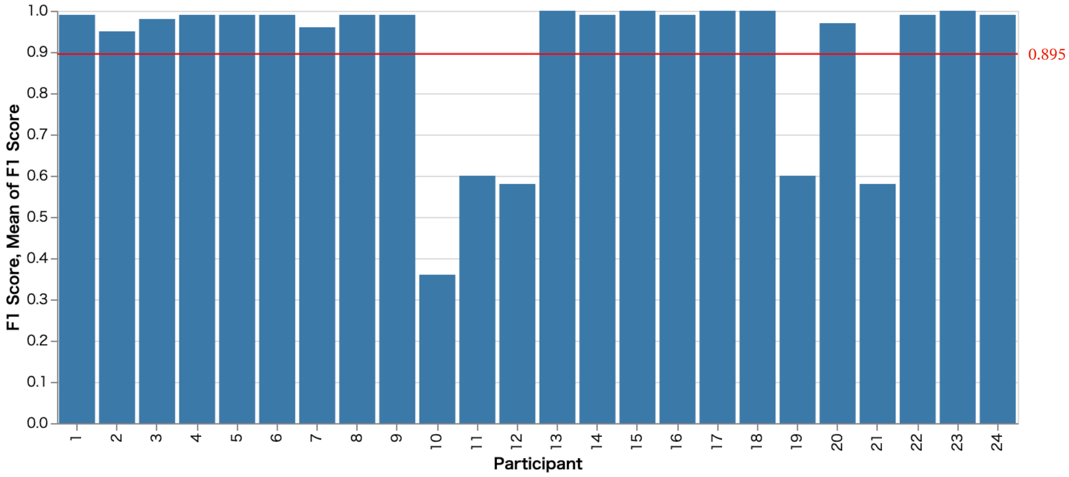


FIGURE 4.13: F1 score of leave one participant out cross validation for MobileNetV2.

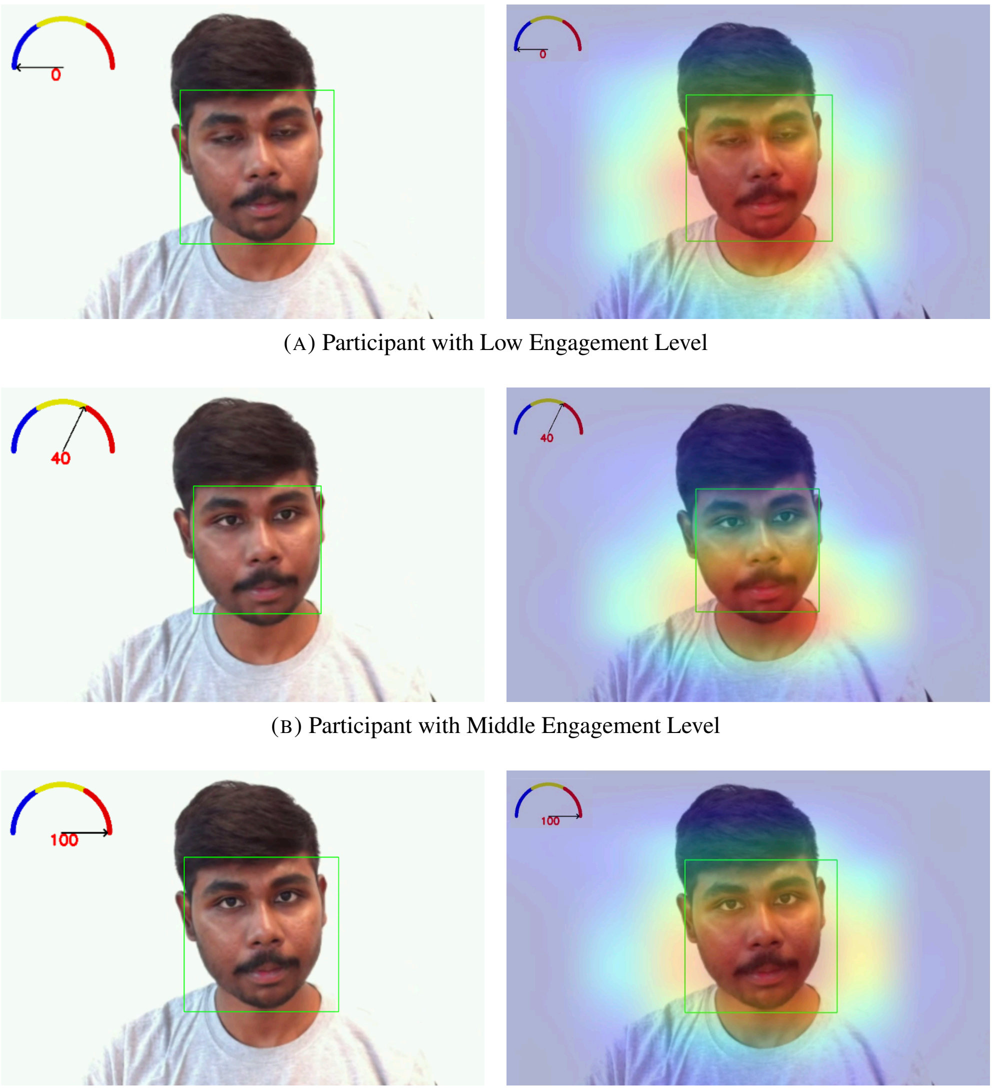
removed as noise in this study. After removing the noise images, the number of images with high, middle, and low engagement levels was 105,960, 149,004, and 105,088, respectively. Figure 4.12 shows the result of the confusion matrix for each transfer-learning model. The average F1 scores of VGG16, Xception, MobileNetV1, and MobileNetV2 were 0.549, 0.905, 0.917, and 0.923, respectively. As a result, MobileNetV2 achieved the highest classification rate in our case study.

Comparison of feature extraction based and deep learning based model

The feature extraction and deep learning comparison will be made by comparing Random Forest and MobileNetV2, which had the highest accuracy for each approach. For the simple comparison, we apply leave-one-group-out cross-validation to get the result of the F1 score. Data from one group (three participants) was used as test data, and the training model consisted of data from the remaining seven groups (21 participants). Table 4.6 shows the result for each recording data. In all cases, MobileNetV2 scored higher accuracy than Random Forest. As a result, MobileNetV2's mean F1 score was 0.923, which is higher than Random Forest's 0.467.

Leave one participant out cross validation of MobileNetV2

We apply the leave one participant out approach to the best-performed MobileNetV2 deep learning model. Figure 4.13 shows all participant results of the F1 score. The highest F1 score was 1.00 for five participants, and the lowest was 0.36. The mean F1 score of leave-one-participant-out cross-validation was 0.895. Compared with the previous research [76], our model's ternary classification rate of engagement levels scored highest. Hence, the performance of the machine learning model is enough to be used for the primary model of the engagement level classification application.



(A) Participant with Low Engagement Level

(B) Participant with Middle Engagement Level

(C) Participant with High Engagement Level

FIGURE 4.14: This is the user interface of the EnGauge system. Engagement levels are displayed as a gauge in the user interface. As the gauge moves to the right, it indicates that the user is engaged at a high level. The left image in each sub-figure shows a raw image and the right image shows a heatmap after applying Grad-CAM [151].

4.2.3 Application

This section explains the end-user application using the engagement recognition model proposed in Section 4.2.2. We first present the user interface of *EnGauge* in Section 4.2.3. Then, we explain the pilot study of how well *EnGauge* can be used in another type of online meeting study in Section 4.2.3.

User interface

The user interface of the application *EnGauge* is presented in Figure 4.14. The model predicts absolute-value percentages with regression of engagement levels from 0 to 100. The left side of the gauge represents low-level engagement. When the gauge bar goes to the right, it infers that the user's engagement level gets high. The gauge consists of three colors: blue,

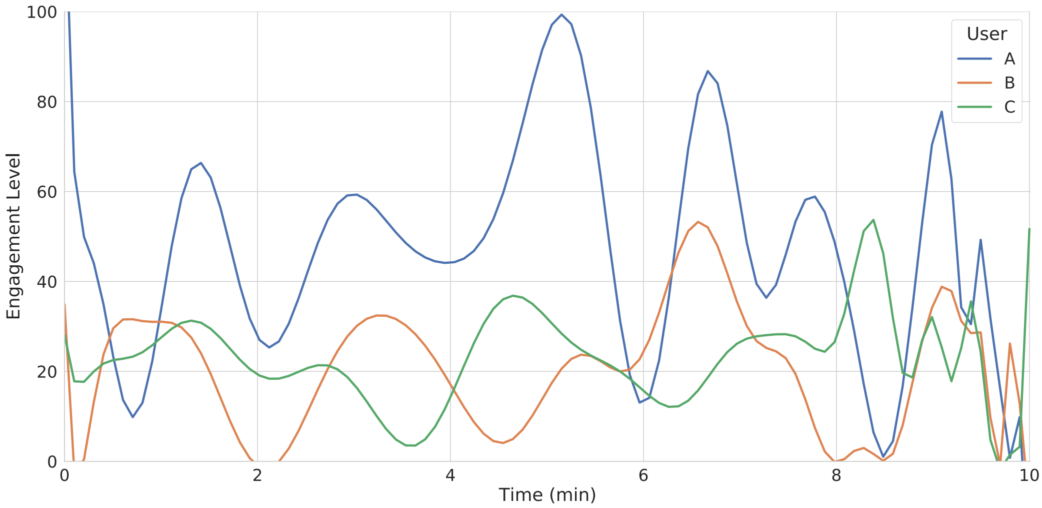


FIGURE 4.15: Changes in engagement level transitions during the pilot study experiment for three participants.

yellow, and red. These colors represent the classification level of low, middle, and high. The advantage of this user interface is that it triggers user motivation to use the system when participating in online meetings to increase the gauge, thus increasing engagement in a game-like manner. Users can benefit from understanding their engagement level using color and numeric percentage information.

We also make the visualization with heatmap using Grad-CAM [151] as explained in Figure 4.14. The heatmap shows the important features inside the input facial image. The red color represents a vital area feature for each engagement detection. On the other hand, the blue area represents an area of feature with a weak weight for detection. As shown in each image, the recognition model recognizes the area around a person's face as important information for classification. The mouth is more important than the eyes, especially in middle engagement level detection.

As a result, the gauge interface can support user engagement as feedback and real-time. Also, the video can be feedback for the online meeting facilitator to understand who was engaged in which scene with what kind of critical behavior.

Pilot study of the system EnGauge

We conduct a pilot study using the *EnGauge* system presented in Section 4.2.3. The pilot study involved three volunteers participating in the Alternative Uses Task (AUT) [62]. Volunteers are all native English speakers, so the experiment is conducted online and speaks in English. In the AUT, participants are asked to list as many alternative uses of a given object (e.g., spoon, hanger, tennis ball) as possible. We used the AUT to ensure all participants were activated during a group meeting. The experiment takes time for ten minutes. Unlike our initial data collection, participants joined the online meeting under their conditions. Hence, the camera resolution or screen background for each user was different.

Figure 4.15 shows the result of a change in engagement level transitions during the pilot study for all three participants. For the data visualization, we apply the method as follows. First, for each second, we get the results of the engagement level as a percentage. Then, we apply a regression model to make the graph curve. Every three participants' engagement level flow is shown in different colors. From the result, the exciting observation was that Participant A frequently changed the engagement level compared with Participants B and C. Participant A often stays in high engagement compared with B or C. When we examined the video in detail, we confirmed that Participant A was becoming like a facilitator who talked to lead the meeting. Another observation was the timing of the engagement level change. Participants A and B or A and C often improve their engagement levels simultaneously. This result can assume that the group has some synchronization in engagement, as mentioned in the previous research [56]. Looking back at the video, we see that this behavior occurs when users discuss or react to one another's opinions. We then discover from this result that external factors can control engagement. An interesting observation from the pilot study was that while two participants increased their engagement levels, another tended to decrease their engagement levels. This happened in the original video because two participants were discussing, and the other participant was left over. Being out of conversation made participants less involved intensely in the online meeting.

4.2.4 Discussion

In this section, we discuss about the contributions. First, we discuss the uniqueness of data collection. Secondly, we discuss our engagement detection model. Lastly, we discuss the engagement detection application *EnGauge* and pilot study.

Uniqueness of the data collection

One of our study's core contributions is the uniqueness of the data collection strategy. We reviewed the literature of previous studies to understand the strategy of collecting a dataset for each engagement level. As it is mentioned in Table 2.1, the data collection for engagement levels is mainly done by using self-annotation [117], observer annotation [117], survey [45], or using real-time feedback from participant [10]. The limitation regarding this labeling was time-consuming, and labeling accuracy decreased when an observer did it. By recruiting each user with high, middle, and low engagement level behavior, we can use all the recording data for each labeling. Also, this way, we can overcome the limitation of becoming too much in-the-wild study mentioned by Gashi *et al.* [58]. The researcher mentioned that a controlled data collection setting ensures a clean dataset. For example, if we let the user do anything in the online meetings, the user could move away from the screen to discuss only with a microphone. Our condition of making a role for each user supports the experiment to be under control.

Another strength of our data collection strategy is the reliability of each engagement-level dataset. DiSalvo *et al.* has mentioned as a limitation of their work that participants tend to make fake engagement [47]. This behavior happens because users are told to focus on their engagement levels. Learning from previous research, we focused not on describing engagement but instead on asking the participant to do a particular activity. Table 4.2.2 mentions that each activity represents each engagement. Therefore, our data collection approach is beneficial in collecting a clean dataset.

Lastly, increasing the number of balanced datasets is one of our strengths. Gao *et al.* mentioned that creating a large corpus of engagement-level labeled dataset collections was the limitation. To overcome this issue, our data collection strategy is effective. Since we asked each participant to behave at certain engagement levels, we collected almost the same amount for all engagement-level datasets. The dataset did not become equal since some participants faded from the screen or did not move for a long time and were removed by gray-scaling hashing. However, as we mentioned in Section 4.2.2, we collected 105,960, 149,004, and 105,088 for each high, middle, and low engagement level. After removing the duplicate images, we confirmed 264,688 for the engagement recognition model. We can confirm that the balance of each engagement dataset was equally collected.

In summary, our data collection strategy contributes to creating a reliable and balanced label dataset corpus by setting the control experiment condition.

Best performed engagement level detection model

One of the core contributions of this study was the highest remark on the prediction rate of engagement level. Our study remarks 0.895 in F1 score with leave one participant out approach and compare to related works it remarks highest as shown in Table 2.1. When we tried the feature extraction-based approach, we could only achieve 0.467 in the F1 score for the leave-one group out cross-validation. The group refers to the recording members, containing three participants each. Hence, the deep learning-based approach was effective in our case study compared with the approach mentioned in previous works. Before discussing the result of the deep learning approach, we first look at the feature extraction-based approach.

An exciting finding in the feature extraction-based approach was that middle and low-engagement participants get confused according to the confusion matrix mentioned in Figure 4.11. This confusion occurs due to the similar actions that occur for each engagement level. Since the feature extraction-based approach uses the vectorized format, it is essential to know what action occurred. We have found that confusion mainly happens when the user is typing on a keyboard. The user's bottom face will be hidden when the engaged middle user looks down to see the keyboard. The user's mouth is essential to discover if the user is speaking, but looking down while typing on the keyboard compromises the mouth information. This behavior only exists for highly engaged users, leading to misclassification between middle and low-engaged users. Overall, the accuracy remark could be higher for the feature-based approach, mainly because the actions are more or less standard for all high, middle, and low engagement situations. Hence, discovering the engagement level from the sliding window approach could have performed better.

The deep learning-based approach confirmed that the F1 score is the highest value. We first applied this model because our previous work [117] applied VGG16. Similar to the previous study, we confirmed the performance with VGG16 with an F1 score of 0.549, higher

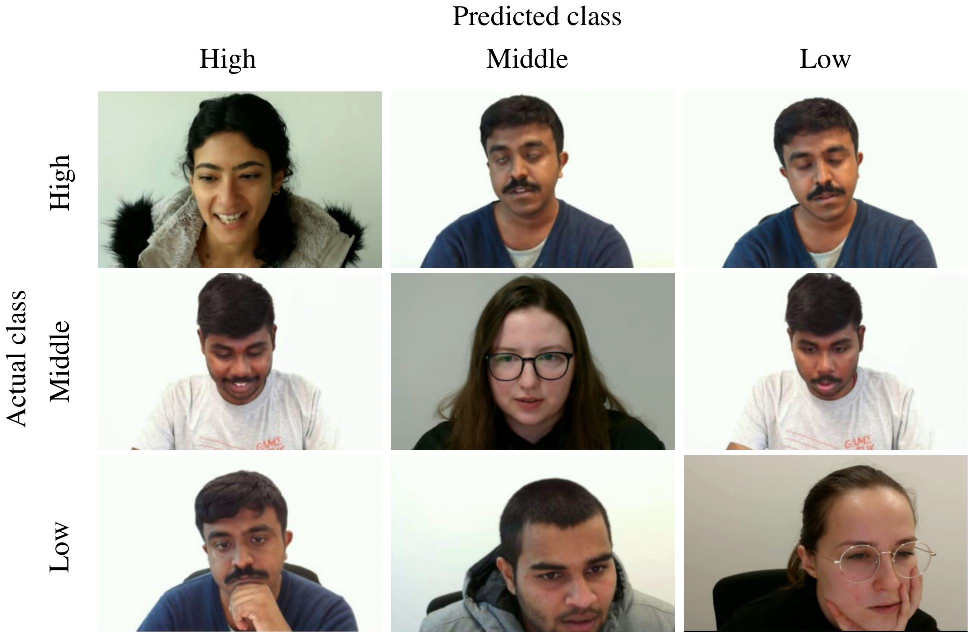


FIGURE 4.16: Outputs of the engagement level classification using MobileNetV2.

than the 0.467 scores calculated with feature extraction-based (Random Forest). However, it is lower than other models such as Xception, MobileNetV1, and MobileNetV2. We discover that MobileNetV2 performed the best, as mentioned in Table 4.2.2. We then applied to leave one participant out of cross-validation for MobileNetV2 as shown in Figure 4.2.2. As a result, we confirmed the mean F1 score of 0.895 for 24 participants. However, we discover the lowest F1 score of 0.36 for one participant. We try to discover the reason for the low accuracy by looking at the original image.

Figure 4.16 shows a confusion matrix with the original image. From the image, we found some possible reasons for misclassification. In the image, we discover that the mouth is one of the important features for predicting engagement. As mentioned in Figure 4.14, we confirmed this with Grad-CAM. When we compared the user with the actual class of middle-level engagement users, participants who were misclassified as high engagement, we found that the user was smiling. It is confirmed that the model predicts that the user had a high engagement due to this characteristic. On the other hand, users who are high engagement for the actual class and misclassified as low were closing their eyes and mouths. This feature tends to be classified as low engagement since none of the significant high engagement characteristics were found.

In summary, we have discovered that the deep learning-based approach, especially MobileNetV2, is best for engagement recognition. The benefit of using MobileNetV2 is that the model can be used without a GPU (graphic processing unit) machine [146]. To run our system, users can use their local computer with a CPU (central processing unit) to recognize their engagement levels. Although we have yet to discover it in our current paper, we assume that our model can also be used for mobile phone cameras. Hence, our implemented dataset and deep learning model will be helpful for many scenes of engagement analysis.

Implementation of the application and conducting a pilot study

One of the core contributions of this study was implementing the engagement level visualization interface using gauge design. Our initial application prototype was presented in Section 4.2.3. In the initial stage, we made an interface with high, middle, or low engagement labels as text in the extracted frame image. When we tested this interface, it took users time to understand the extent to which their engagement was improving dynamically. Then, we develop a gauge user interface for the engagement level visualization.

The gauge-like user interface encourages the user to understand their level of engagement instantaneously. Since the gauge has blue, yellow, and red, each user will try to make the gauge into a red area. It encourages users to increase their gauge and engage in online meetings. We also show the percentage of the engagement level numerically. The numerical information helped the users understand if they had reached the maximum level of engagement.

After implementing the gauge-like user interface of *EnGauge*, we conduct a pilot study. From the pilot study, we found that engagement levels tend to increase synchronously between two participants. In an online meeting with three participants, one person may be left out because two participants are often interacting with each other. Specifically, it refers to a situation in which one participant is the sender and asks some question, a second participant is the receiver and answers the question, and a third participant observes the two on the sidelines. Interestingly, we saw a proportional behavior of the engagement transition results between the two in Figure 4.15. After four minutes of the meeting, User B's engagement level increased, and we also observed a proportional increase in engagement level for User A. When we look at the actual video, we confirm that User B asked User A some questions/opinions. The behavior of asking questions and opinions encouraged two users to improve their engagement levels.

We also discovered findings on the importance of engagement. When we checked the user performance of AUT result, we found that the engagement level does correlate. The number of ideas proposed by each participant, A, B, and C, was 22, 14, and 12. The highly engaged users were in the same order: A, B, and C. The results indicate that it is feasible to enhance individual participants' engagement levels, which would increase their contribution.

In summary, we have confirmed that the application we have developed, *EnGauge*, has a high potential to understand and improve the transition of user engagement in online meetings. From the pilot study, we have also identified that improving engagement can improve the user's performance.

Limitations and Future work

The work's contributions were strongly mentioned; however, some limitations exist for future research to refine and extend our contributions.

The first limitation is the guidance of the control setting or conditions of the online meeting experiment. In our experiment, we did not guide participants to make their faces out of the camera frame. Since then, we have found some imbalances in each dataset level.

Specifically, we found 149,004 as a usable image for the middle engagement labeled dataset. However, 105,088 was usable for the low-engagement labeled dataset. The difference in the number of images for the middle and low-engagement labeled datasets was more than 40,000 frame images. We found that most of our image reduction occurred mainly due to the unclear detection of a user's face. In detail, some participants lost more than 20% of the frames. The chin, for example, was outside the box and was not captured by the camera. We could remind each user to be inside the camera's frame for future work. Another option is to notify the user when they are out of the camera frame during the experiment.

The second limitation is the redundancy of the dataset. Our most significant contribution was to create a controlled environment for the experiment to create a clean dataset. Due to these settings, we have some limitations in the behaviors that could occur in the wild. We asked participants to take their text files on the computer during the experiment. This guidance significantly avoided looking at notes or documents outside the computer. However, meeting participants can use paper materials to take notes in real-life situations. We also prevent users from speaking to someone outside the screen. This behavior could also happen in wild studies. Therefore, since the conditions of this study are not entirely wild, various cognitive errors may occur when adapted to the real world. We could discover these unknown behaviors in the wild and apply those patterns to a trained model for future work. Our next target is to continuously integrate the model while the user is using *EnGauge* application.

The third limitation is the variance of belongings for each participant. We only asked participants to remove their facial masks during our data collection before starting the experiment. However, we found that participants' belongings varied after completing the data collection. Examples of the belongings are glass, cap, or hijab during the experiment. These accessories tend to hide the faces of the participants. We should fix participants' belongings to make the experiment conditions more controlled and generate a clean dataset. For future work, we could make a dataset category. Then, we could implement a model for each category to generate an accurate model.

The fourth limitation is the variance of the participants' backgrounds. Similar to the third limitation, we realized that our dataset collection has a variety of backgrounds. The background refers to the nationality, age, gender, or relationships with other experiment participants. Other than the last condition, nationality, age, and gender somehow correlate with the behaviors. It is often mentioned that differences in cultural backgrounds may have different ways of expression or communications [103, 109]. For future work, we could also classify the backgrounds of participants into some categories to make the model more precise.

The fifth limitation is the improvement of the engagement level recognition model. As we mentioned, the model can still be improved due to the participants' different backgrounds. In our current MobileNetV2 model, we only put an image frame into the model for recognition. The model tries to recognize the information in the image according to the hashed information. However, we could add additional information, such as the emotion recognition model [64]. For future work, we could extend our model and create an emotion recognition model to make a more robust engagement recognition model.

The last limitation is the user interface of *EnGauge*. Our initial user interface allows the user to see the gauge on the screen during online meetings. Also, users can get feedback afterward by recording it if necessary. However, the current version of showing the gauge may distract the participant from the conversation since the user might often look at the gauge. Our target is not to attract the user's attention to the gauge interface but to engage the user in the meetings. In order to do so, the application can be in another kind of approach. In future work, we could apply our engagement levels detection model to various applications. For example, implement a meeting facilitation Bot to check each participant's engagement level and ensure everyone is on the same page. Another idea is to create a notification that users can receive when their engagement level has been low for a certain amount of time.

4.2.5 Conclusion

This work uses only a built-in webcam to measure student engagement levels during an online meeting. Our approach can generally achieve the ternary classification of the engagement levels. We performed our analysis on data collected from 24 students after preprocessing. We first apply a feature extraction-based approach and remark an accuracy of 46.7%. Then, we applied a deep learning-based approach, MobileNetV2, achieving an average F1 score of 89.5Using our model, we developed an application called *EnGauge* and conducted the pilot study. The application has a high potential to understand the transition of human engagement in an online meeting. Our contribution demonstrates a new data collection approach, an optimal engagement level recognition model, and application scenarios.

4.3 Concern Gender as a Feature for Emotion Detection

In human-computer interactions (HCI), a camera can be defined as an eye for the computer to see humans. Computer eyes play a role significantly in various areas, such as understanding humans' body movements [25, 34], hand gestures [38, 125], or cognitive states [5, 10, 175]. Previous research suggested that gender differs in emotional expression [32, 96]. Emotions are the heartbeat of our daily lives, intricately entwined with how we interact with the world. Lately, the scientific world has been growing fascinated with how we express and recognize emotions. This becomes even more intriguing when we consider how gender influences these emotional landscapes, it is a complex dance between psychology, neuroscience, and the burgeoning field of artificial intelligence. With the leap forward in technology, especially neural networks and machine learning, we have opened a new chapter in understanding emotions. These advancements are not just remarkable tech developments; they offer a deep dive into the subtle differences in how different genders experience and express emotions. However, as exciting as this is, it has its challenges. When we mix gender, emotion, and technology, we face tough questions about biases and blind spots in our current practices.

Our journey through this research explores the complex interplay of gender and emotions, combining insights from gender studies with advancements in neural network technology. Our primary mission is to unravel how gender influences emotional expressions and to develop predictive models that accurately represent these nuances across all genders. This work is more than just a theoretical inquiry; it is a venture into creating equitable and functional technology in diverse real-world settings. We aim to transform our understanding of emotional dynamics across genders, paving the way for applications that enhance human-computer interactions, enrich educational experiences, and refine communication strategies in various professional and social contexts. This paper draws inspiration from some of the best minds in gender studies, emotional psychology, and explainable AI. We dissect how different genders show emotions using a treasure trove of data and innovative methods.

In this paper, we state two research questions. (1) Does gender as a feature improve the accuracy of predicting human emotion using a camera? (2) How do facial emotions differ between genders? Our work does not just add to the conversation about gender differences in emotional expression but also makes our predictive models more inclusive and accurate. Ultimately, we are working towards technology that effectively and ethically understands and serves everyone, regardless of gender.

4.3.1 Dataset

In this work, we use the open-sourced dataset DAiSEE [63]. The DAiSEE dataset is a meticulously designed resource for studying learner emotions in e-learning environments.

Data Collection and Annotation

One distinguishing feature of the DAiSEE dataset is its commitment to realism and authenticity in data collection. High-definition web cameras were used to record videos, ensuring

TABLE 4.7: Composition of Frames by Emotion and Gender in the DAiSEE Dataset

Emotion	Total Images	Male Images	Female Images
Boredom	42,121	17,180	24,941
Confusion	27,818	8,370	19,448
Frustration	18,062	6,235	11,827
Engagement	71,100	30,910	40,190

visual clarity and quality. Notably, the dataset was collected in natural settings where learners use e-learning materials, including dorm rooms and libraries. This approach captures learners' genuine reactions and introduces variations in illumination conditions, enhancing the dataset's realism.

The annotation process of the DAiSEE dataset is thorough and reliable. Initial annotations are crowdsourced, incorporating diverse perspectives on interpreting learner affective states. This diversity mirrors the potential variations in how viewers perceive a learner's engagement. Expert psychologists established a gold standard to validate the accuracy and reliability of these annotations. This gold standard is a benchmark, ensuring that the crowdsourced annotations meet high accuracy and consistency. The correlation between the crowdsourced annotations and the expert-created gold standard is pivotal in refining the data, eliminating biases, and enhancing the overall quality. Moreover, the DAiSEE dataset utilizes the Dawid-Skene aggregation algorithm for vote aggregation. This statistical model effectively handles data from multiple annotators, especially in scenarios with no definitive ground truth. It accurately aggregates varied annotations into a consistent set of labels representing the intensity and type of learner engagement. The DAiSEE dataset provides a rich and robust resource for understanding and analyzing learner engagement in e-learning environments through its comprehensive content, detailed multi-level labeling, and rigorous annotation process. Its composition reflects the complexity of human emotions and behaviors, while its methodology ensures the authenticity and reliability of the data, setting a high standard in affective computing research.

Data Composition

The DAiSEE dataset comprises 9068 video clips from 112 diverse users. Table 4.7 illustrates the dataset composition. This diverse pool of users ensures that the dataset reflects a broad range of user interactions and responses in e-learning settings, making it highly representative. Central to the dataset's utility is its multi-label video classification scheme, which categorizes user affective states into four primary categories: boredom, confusion, engagement, and frustration. Each category is divided into four intensity levels, allowing for a precise assessment of learner engagement and emotions. To provide a deeper insight into the dataset, we have conducted a frame extraction process from each video, resulting in a detailed breakdown by emotions and gender.

TABLE 4.8: Summary of Convolutional Neural Network Models

Model	Key Attributes
MobileNetV2	- High computational efficiency - Adapted for 224x224x3 input images - Includes a 1024-unit Dense layer with ReLU activation and a Dropout layer - Concludes with a 4-neuron sigmoid-activated output layer
InceptionV3	- Depth and efficiency for 299x299x3 input images - Similar layer structure with Dense and Dropout layers - Ends with a 4-neuron output layer
Xception	- Focuses on depth in processing - Configured similarly to InceptionV3 in terms of input size and layer setup
VGG-Face	- Specializes in facial recognition - Designed for 224x224x3 inputs - Sequence of GlobalAveragePooling2D layer, Dense layer, and Dropout layer - Ends with a 4-neuron output layer with sigmoid activation

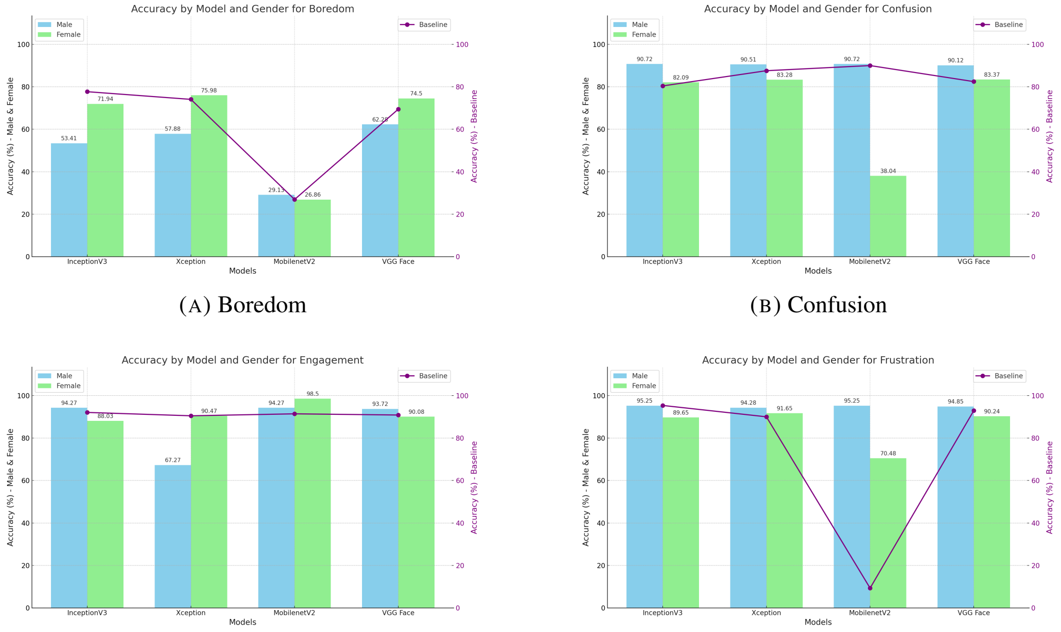
4.3.2 Methodology

Building upon the comprehensive foundation provided by the DAiSEE dataset, our study ventures into deep learning to estimate learner engagement through video data analysis. We employed four advanced convolutional neural network (CNN) models, each selected for its distinct capabilities in image processing and pattern recognition: MobileNetV2 [146], InceptionV3 [167], Xception [37], and VGG-Face [130]. These models' key attributes and configurations are detailed in Table 4.8. Each model underwent meticulous data preparation, including preprocessing and splitting the DAiSEE dataset into training, validation, and testing sets (60-20-20 ratio). Standard image preprocessing techniques were applied to conform to the models' input specifications. The training was conducted with a batch size of 256 over 30 epochs, with early stopping implemented to prevent overfitting. Model performance was evaluated using accuracy metrics and confusion matrices, providing detailed insights into classification effectiveness.

In our study, a critical component involves comparing two approaches applied to the deep learning models: the baseline methodology and the advanced strategy combining data augmentation with the leave-one-gender-out approach. This comparative analysis is pivotal in understanding the robustness and adaptability of the models under different training conditions.

Baseline Methodology

All four models, MobileNetV2, InceptionV3, Xception, and VGG-Face, were initially trained and evaluated using a standard approach. This baseline methodology involved training the



(A) Boredom

(B) Confusion

(C) Engagement

(D) Frustration

FIGURE 4.17: Result of the emotion recognition with baseline, male-only, female-only.

models on the complete dataset without gender-specific exclusions or data augmentation techniques. It served as our control setup, providing a benchmark for the models' inherent capabilities in classifying emotional states from video data.

Data Augmentation and Leave-One-Gender-Out Approach

In contrast to the baseline, the second approach introduced data augmentation and the leave-one-gender-out strategy. Here, the models were trained on datasets excluding one gender at a time, challenging them to generalize across more diverse datasets. Data augmentation techniques, such as random adjustments in image brightness and contrast, were employed to enhance further the models' ability to adapt to variations in the data. This approach tested the models' robustness and performance in less biased and more generalized settings.

The performance of each model under these two approaches was meticulously compared using a range of metrics, including accuracy, precision, and recall. This comparison aims to reveal the impact of data augmentation and gender-specific training on the models' ability to classify emotional states accurately. The detailed results and thorough analysis of this comparative study are presented in the following section, "Results." This section will delve into the specifics of each model's performance under the two approaches, providing insights into their strengths and limitations.

4.3.3 Results

The results from the analysis of four different models demonstrate varied performances across different emotions and gender categories. Detailed accuracy by model and gender for each emotion is illustrated in Figure 4.17. InceptionV3 exhibited notable proficiency in recognizing 'Engagement' with high accuracy, particularly in males (94.27% accuracy). However, it struggled significantly to identify 'Boredom' in males (53.41% accuracy) and females (71.94% accuracy). 'Confusion' and 'Frustration' detection was strong in males but showed a stark contrast in females. These patterns are visually depicted in Figures 4.17c 4.17a, 4.17b, and 4.17d for 'Engagement', 'Boredom', 'Confusion', and 'Frustration'.

Xception presented a mixed performance, excelling in 'Engagement' detection, especially in females (90.47% accuracy). The model showed moderate effectiveness in recognizing 'Boredom', with slightly better female performance. For 'Frustration' and 'Confusion', the model demonstrated high accuracy for males and relatively lower but good accuracy for females.

MobileNetV2 demonstrated high effectiveness in identifying 'Engagement', particularly in females (98.50% accuracy). The accuracy for 'Boredom' was considerably low for both genders. However, the model performed well in detecting 'confusion,' with high accuracy in males (90.72%). However, the accuracy dropped significantly in females (38.04%). The model showed high accuracy in recognizing 'Frustration' in males (95.25%) but lower accuracy in females (70.48%).

VGG-Face showed consistent performance across all emotions, with high accuracy in detecting 'Engagement' and 'Frustration' across genders. It had moderate success in recognizing 'Boredom', with better accuracy in females (74.50%). The detection of 'Confusion' was more effective in males than females.

In the comparative analysis between the baseline analysis and the leave-one-gender-out approach with data augmentation for the MobileNetV2, InceptionV3, Xception, and VGG-Face models, distinct trends and disparities emerge, highlighting the impact of gender-specific data handling in emotion recognition tasks.

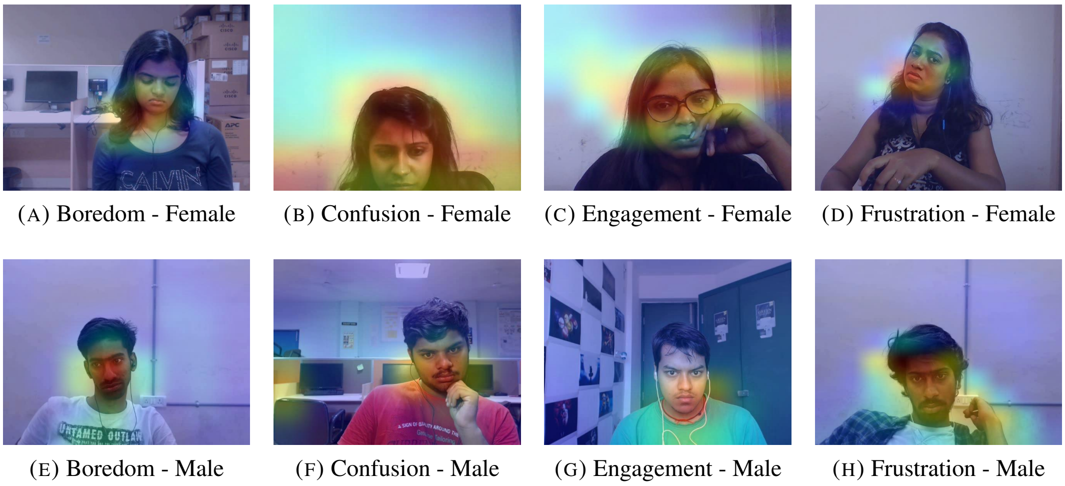
InceptionV3 exhibited notable performance differences. The gender-specific analysis maintained high accuracy for 'Engagement' at 88.03% for females, and showed improvements in 'Boredom' and 'Confusion' for females, but no significant change in 'Frustration'.

Xception's performance in 'Engagement' was consistent at 90.47% across both analyses, with minor accuracy variations in 'Boredom' and 'Confusion', and stability in 'Frustration'.

MobileNetV2 achieved a strong baseline in 'Engagement' at 91.39%, with increased accuracy for females in the gender-specific approach reaching 98.50%. However, it showed low performance in 'Boredom' across both approaches, decreased 'Confusion' accuracy for females in the gender-specific approach, and a significant increase in 'Frustration' accuracy for females, highlighting the impact of gender-specific data augmentation.

VGG Face demonstrated high baseline accuracy in 'Engagement' at 90.83% and maintained strong performance for females in the gender-specific approach. It showed a notable increase in accuracy for 'Boredom' and 'Frustration' in females and a slight decrease in 'Confusion' accuracy, but overall robust performance in the gender-specific analysis.

Additionally, the application of Grad-CAM technology provided insightful visualizations, revealing notable differences in how emotions are expressed between genders and highlighting key facial features vital in emotion recognition.



(A) Boredom - Female (B) Confusion - Female (C) Engagement - Female (D) Frustration - Female

(E) Boredom - Male (F) Confusion - Male (G) Engagement - Male (H) Frustration - Male

FIGURE 4.18: Grad-CAM visualizations showing gender-based distinctions in facial regions for different emotions.

4.3.4 Discussion

Our research illuminates gender differences in emotional expression and recognition, emphasizing their relevance to the development of technology for human-computer interaction (HCI) and e-learning. Aligning with existing literature on gender-based emotional variance, our study offers practical insights for enhancing emotion recognition technologies by incorporating male-female differences. We integrated gender-specific data into different convolutional neural network models and found significant improvements in recognizing emotions like 'Engagement', with varied results between genders. However, we also noted that these models struggle to consistently identify emotions such as 'Boredom' and 'Confusion' across genders. These findings underscore the importance of tailoring technology design to accommodate gender differences.

The comparative analysis sheds light on how gender-specific data handling impacts emotion recognition. InceptionV3, for example, showed improved accuracies in detecting certain emotions in females under the gender-specific approach compared to the baseline. Xception maintained steady accuracy in 'Engagement' detection across both methods. MobileNetV2 saw a notable increase in accuracy for 'Engagement' in females in the gender-specific approach, while VGG Face demonstrated slight variations in accuracy for different emotions across the two methods.

Furthering this analysis, the application of Grad-CAM in our research has been fundamental in highlighting these aspects. The Grad-CAM visualizations bring the specific facial regions essential in differentiating emotions across genders. Our analysis using Gradient-weighted Class Activation Mapping (Grad-CAM) revealed distinct gender-based patterns in the models' recognition of facial expressions.

Boredom - For females, the models focused on the lower half of the face, particularly around the mouth, suggesting that female expressions of boredom might be conveyed through mouth movements or positions (Figure 4.18a). In males, attention was directed to the upper facial region, especially around the eyes and nose, indicating that male expressions of boredom could involve reduced eye movement or a lack of facial engagement (Figure 4.18e).

Frustration - The models highlighted the brow and jawline in males, implying that male frustration might be expressed through tension in these facial muscles (Figure 4.18h). For females, the focus was on the mouth area, suggesting that signs of frustration in females are more evident in movements or expressions related to the mouth (Figure 4.18d).

Engagement - In male subjects, the models concentrated on the mouth and lower nose area, indicating these regions are key in conveying male engagement, potentially through mouth movements or reactions (Figure 4.18g). The eye region was emphasized for females, implying that female engagement is communicated through the eyes, possibly via gaze direction or eye movement (Figure 4.18c).

Confusion - The models focused on the mouth region in males, suggesting that expressions of confusion might be conveyed through mouth movements (Figure 4.18f). In females, the emphasis was on the eye area, highlighting the eyes as key indicators of confusion, possibly manifested through widened eyes or a searching gaze (Figure 4.18b).

These findings underscore the nuanced differences in how emotions are expressed and perceived across genders, providing valuable insights into the facial features most instrumental in emotion recognition.

4.3.5 Limitation and Future work

Our study marks an essential step in emotion recognition, but it is important to recognize its limitations. The primary limitation stems from our reliance on the DAiSEE dataset, which, while comprehensive for e-learning environments, might only partially represent the wide range of emotions in different real-world settings. This limitation impacts the broader applicability of our findings. We also noted differences in the performance of various convolutional neural network models, such as MobileNetV2, InceptionV3, Xception, and VGG-Face, in emotion detection. These differences suggest potential biases, which could influence their effectiveness in varied contexts. Addressing these biases is crucial for improving model accuracy and reliability.

Future research should broaden the scope by including more diverse datasets that reflect a more comprehensive range of emotional states and settings. This would enhance the versatility and robustness of emotion recognition models. Additionally, it is important to analyze and mitigate any inherent biases in these models, ensuring their accuracy and fairness. Another key area for future exploration is the inclusion of a cross-cultural perspective. Recognizing and accommodating the diversity in emotional expression and gender perception across cultures is essential. Future studies should assess how these models perform across different cultural backgrounds to develop more universally relevant and culturally aware technologies.

By addressing these limitations and exploring these future directions, we can make significant strides in creating more reliable and socially relevant emotion recognition technologies. This approach will help develop systems finely attuned to the subtleties of human emotion and expression.

4.3.6 Conclusion

This study explored the intersection of gender and emotion recognition using convolutional neural network models like MobileNetV2, InceptionV3, Xception, and VGG-Face. Key findings demonstrate that incorporating gender-specific data significantly enhances emotion recognition accuracy, particularly for emotions like Engagement and Frustration. This underscores the need for gender-inclusive approaches in technology design, moving beyond traditional gender-neutral methods. Our use of Grad-CAM technology further highlighted the importance of gender considerations in accurately identifying emotional states, marking a significant contribution to affective computing and human-computer interaction. This research paves the way for developing technologies that more accurately reflect the diversity and complexity of human emotions and expressions. In conclusion, the study advocates for developing technologically advanced yet socially conscious and inclusive systems. It opens new paths for future research into gender-inclusive models, aiming to create equitable and effective emotion recognition technologies that resonate with a wide spectrum of human experiences.

4.4 Discover Knowledge Receiver: Comprehension Level Estimation

In the wake of the devastating COVID-19 pandemic, the field of education has undergone a profound transformation, with a rapid shift towards hybrid learning models. Video lectures have emerged as a powerful tool for disseminating knowledge to a vast audience asynchronously. However, the lack of real-time monitoring capabilities poses a significant challenge, leading to an urgent need for a deeper understanding of non-verbal cues from remote audiences, surpassing the traditional scope of in-person lectures. Compelling evidence from previous research underscores the limited attention span of audiences, with studies suggesting a mere 10-20 minutes of sustained focus before attention wanes [23]. Moreover, dropout rates have been found to escalate proportionately to the duration of video content [92]. This alarming trend underscores the criticality of real-time comprehension assessment and continual feedback mechanisms for fostering sustainable, high-quality education.

Integrating timely feedback mechanisms holds immense potential, benefiting educators and learners. Educators can leverage insights from student comprehension feedback to refine their instructional approaches. Simultaneously, personalized interventions targeted at students grappling with comprehension challenges can be instrumental in stemming the tide of dropouts and fostering a supportive learning environment. Learners can also get feedback from the system to receive a summary report on the low comprehension segment.

In this context, our study draws inspiration from the groundbreaking work of Srivastava, which pioneered contactless sensors to gauge learning difficulties in digital learning environments. While their research demonstrated the impact of lecture format on self-reported difficulty levels, it needed to analyze biometric markers to gauge comprehension levels. Notably, the absence of real-time data collection and the potential influence of the self-reporting process on participants' multitasking capabilities pose critical limitations to the study's scope and applicability in real-world settings.

Building upon the foundation laid by prior research, our study integrates comprehensive eye-tracking data, post-lecture self-reported confidence annotations, and post-problem-solving scores obtained during video lectures. By merging self-reported confidence annotations with problem-solving performance, we delineate a nuanced and holistic framework for assessing comprehension states. Our study's findings make significant strides toward addressing the critical gaps in existing research and pave the way for a more nuanced understanding of the multifaceted dynamics underlying remote learning environments. Our contributions to this paper are as follows. (1) Public Dataset: We implement a new gaze, confidence, and comprehension state annotation dataset. (2) Confidence Estimation: We investigate the prediction of self-reported *confidence* state from the dataset. (3) Comprehension Estimation: We investigate the prediction of *comprehension* state from the dataset.

4.4.1 Experimental Design

This section explains the participants' backgrounds and the experiment procedure. The dataset will be publicly available.

TABLE 4.9: An example of the segmentation annotations created by the authors. One round is split into three segments.

Location	Amount	Gender	Age (Mean)	Major of the study in the university
Germany	10 Participants	6M & 4F	24 - 38 (27.3)	7 Computer Science, 2 Psychology, 1 Physics
Japan	10 Participants	8M & 2F	23 - 25 (23.5)	10 Computer Science
Total	20 Participants	14M & 6F	23 - 38 (25.3)	17 Computer Science, 2 Psychology, 1 Physics

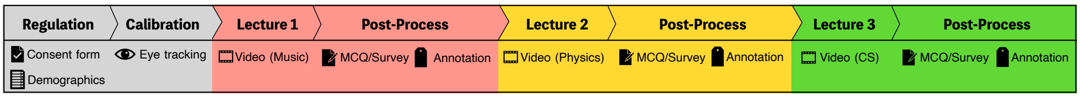


FIGURE 4.19: A figure shows a flow of the experiment procedure. CS in lecture 3 stands for computer science.

Participants

In this study, we recruited 20 participants in Germany and Japan. Table 4.9 shows the detailed background of the participants. We recruit participants in Germany who use English as their main language at university or in their company. For participants in Japan, we recruit those who study or work mainly in Japanese. Against participants in Germany, we obtained participants' consent for the General Data Protection Regulation (GDPR) before the experiment. Participants in both countries could opt out of the experiment at any time.

Procedure

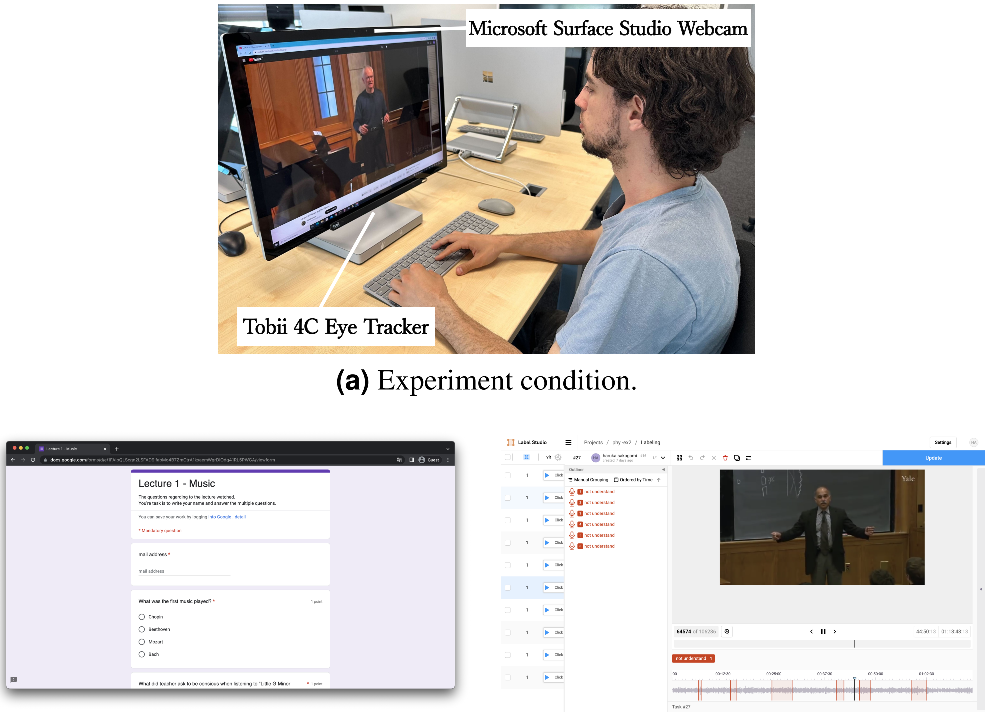
Figure 4.19 shows the overall experiment workflow for participants to follow. Figure 4.20 shows the experimental condition and post-process. In this study, we conduct experiments in the same room in a controlled manner in each country. Participants get an explanation of the experiment's purpose and the use of the collected data. Once they confirm, participants sign the consent form and write demographic information such as gender, age, and major of study. Then, participants do a calibration on the eye-tracker. After calibration, the participant starts watching a lecture video. Once the participant finishes watching, follow the post-process, which is solving Multiple Choice Questions (MCQ), writing a survey, and making a segment annotation of when the user is *unconfident* in the video lecture. Participants repeat this procedure with two more video lectures. To avoid fatigue from the continuous trials, we asked participants to watch each video on different days. The domains of the lectures are music, physics, and computer science.

Materials

We prepared six lecture videos, three types for participants in Germany and another three for Japan. We prepared a lecture in English for participants in Germany and a lecture in Japanese for participants in Japan. Participants in Germany watched the following: music lecture (Lecture 1) [3], physics lecture (Lecture 2) [4], and computer Science lecture (Lecture

[3] https://youtu.be/uDVr0GaD7gI

[4] https://youtu.be/uK2eFv7ne_Q



Lecture 1 - Music

docs.google.com/forms/d/e/1FAIpQLScgn2LSFAD9lfabMo4B7ZmCtrA1kxaemWgrDIOdq41RL5PWGA/viewform

The questions regarding to the lecture watched.
You're task is to write your name and answer the multiple questions.

You can save your work by logging into Google . detail

* Mandatory question

mail address *

mail address

What was the first music played? * 1 point

○ Chopin
○ Beethoven
○ Mozart
○ Bach

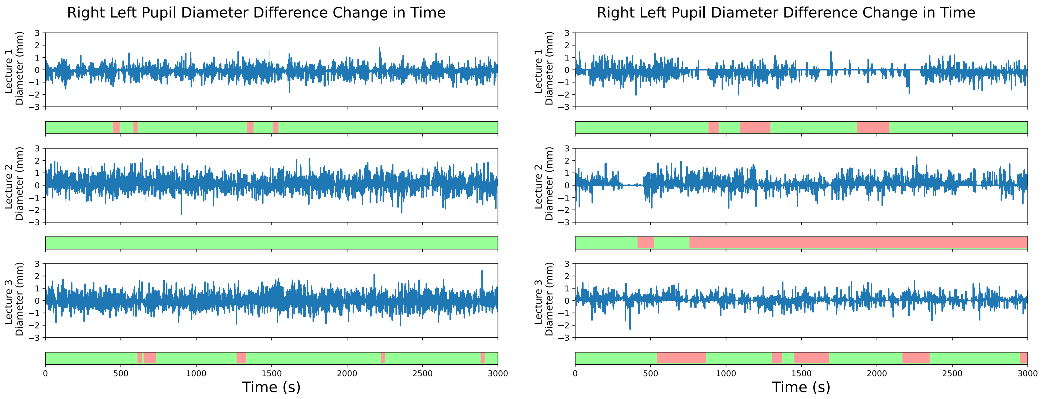
What did teacher ask to be consious when listening to "Little G Minor * 1 point

(a) Experiment condition.

(b) Participant answer MCQ and survey.

(c) Self annotation using Label Studio.

FIGURE 4.20: Experiment condition and post-process. The Tobii 4C eye tracker collects gaze data, and the Windows Surface Studio webcam collects recordings of participants watching a video. Participants answer MCQ/Survey and self post-annotation of *unconfident* time segment is labeled.



(a) Participant 5

(b) Participant 8

FIGURE 4.21: Sample of different participants gaze data (right and left pupil diameter difference) with self-reported confidence annotation. The green rectangular parts represents confident segment and the red rectangular parts represents unconfident segment. As the sample shows, there are significant difference in the count of self-report confidence annotations.

3) [5]. Participants in Japan watched the following music lecture (Lecture 4) [6], physics lecture (Lecture 5) [7], and computer Science lecture (Lecture 6) [8].

[5] https://youtu.be/xv0MnQhVWjI
[6] https://youtu.be/4ZeyWopr1dE
[7] https://youtu.be/jwQY0vOAiOQ
[8] https://youtu.be/-j1hoCubiyE

TABLE 4.10: Comparison of participants' self *unconfident* annotations and question-solving result for participants in Germany.

Participant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Lecture 1 (Music) - Q1	TN	TN	TP	TP	TP	TP	TP	TP	TP	TP
Lecture 1 (Music) - Q2	FN	TN	TP	FN	TP	TP	TP	TN	TN	TP
Lecture 1 (Music) - Q3	FN	FP	TP	FN	FP	TP	TN	TP	TP	TN
Lecture 1 (Music) - Q4	TP	TN	TP	FN	TP	FP	FN	FP	TP	TP
Lecture 1 (Music) - Q5	FN	TN	TP	FN	TP	FP	TP	FP	TP	FP
Lecture 1 (Music) - Q6	TN	FP	TP	TN	FP	TP	TP	TP	TP	TP
Lecture 1 (Music) - Q7	FN	TN	TP	TP	TP	TP	TP	TP	FN	FP
Lecture 1 (Music) - Q8	FP	FN	TP	FN	TP	TP	TP	TN	TP	TP
Lecture 1 (Music) - Q9	TN	TN	FN	FN	TP	TN	TN	TP	TN	TN
Lecture 1 (Music) - Q10	TP	TN	TP	TP	TP	TP	TP	TP	TP	TP
Min continuous *unconfident* time (s)	19.7	135.2	9.4	49.4	32.9	42.8	23.0	72.6	39.5	12.6
Max continuous *unconfident* time (s)	320.0	428.9	18.6	257.3	49.4	89.1	52.7	207.9	72.6	127.3
Mean continuous *unconfident* time (s)	148.9	273.8	14.0	101.3	43.7	70.4	33.8	167.2	56.0	47.4
Total *unconfident* time (s)	1042.5	821.5	28.0	1115.2	174.8	221.2	372.8	501.7	168.2	379.3
Lecture 2 (Physics) - Q1	TP	TP	TP	TN	TN	TP	TP	TP	TP	TP
Lecture 2 (Physics) - Q2	TP	TP	TP	TN	TP	TP	TP	TP	TP	FN
Lecture 2 (Physics) - Q3	TP	TP	TP	FP	TP	TP	TP	TP	TP	TP
Lecture 2 (Physics) - Q4	FN	FN	TP	FP	TP	TP	TP	TP	FN	TP
Lecture 2 (Physics) - Q5	FP	TN	TN	FN	TN	TP	TP	FN	TN	TP
Lecture 2 (Physics) - Q6	FP	FN	TN	FN	TP	TN	TP	FP	FN	TP
Lecture 2 (Physics) - Q7	FN	TN	TP	FN	TP	TP	TP	FP	TP	TP
Lecture 2 (Physics) - Q8	FP	FP	TP	FP	TP	TP	TP	FP	TN	TP
Lecture 2 (Physics) - Q9	TP	FP	TP	FN	TP	TP	FN	FN	FN	TN
Lecture 2 (Physics) - Q10	FN	FN	TP	TP	TP	TP	TP	FP	TN	TP
Min continuous *unconfident* time (s)	14.0	226.3	28.2	89.6	0.0	28.2	15.1	113.1	56.5	12.6
Max continuous *unconfident* time (s)	374.4	679.1	33.0	1234.1	0.0	84.8	67.1	2245.3	641.4	228.5
Mean continuous *unconfident* time (s)	128.0	343.3	29.7	727.4	0.0	58.0	112.7	965.2	231.0	47.4
Total *unconfident* time (s)	1271.1	1259.6	146.2	2362.5	0.0	240.5	280.0	2358.5	1386.5	703.0
Lecture 3 (Computer Science) - Q1	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
Lecture 3 (Computer Science) - Q2	TP	TP	TP	TP	TP	TP	TN	TN	TP	TP
Lecture 3 (Computer Science) - Q3	TP	TP	TN	FN	TP	TP	TP	FN	TP	TP
Lecture 3 (Computer Science) - Q4	TP	TP	TP	TP	TP	TP	TP	TN	TP	TP
Lecture 3 (Computer Science) - Q5	TP	TP	TP	TN	TP	TP	TP	TN	TP	TP
Lecture 3 (Computer Science) - Q6	TP	TP	TP	FN	TP	TP	TP	FP	TP	TP
Lecture 3 (Computer Science) - Q7	TP	TP	TP	FP	TP	TN	TP	TP	FN	FN
Lecture 3 (Computer Science) - Q8	TP	TP	TP	TP	TP	TN	TP	TP	TP	TN
Lecture 3 (Computer Science) - Q9	TN	TN	TN	TP	TP	TP	TP	FN	TN	TP
Lecture 3 (Computer Science) - Q10	TN	TN	TN	FN	FN	FP	FN	FN	TP	TP
Min continuous *unconfident* time (s)	4.5	27.1	5.5	369.8	31.7	33.2	33.1	53.8	49.8	47.1
Max continuous *unconfident* time (s)	39.5	63.4	35.9	1234.1	81.5	116.0	44.1	330.8	176.7	194.4
Mean continuous *unconfident* time (s)	20.9	44.4	18.2	975.2	52.4	226.5	39.6	222.0	74.3	95.3
Total *unconfident* time (s)	110.1	185.8	103.9	1669.6	249.2	198.7	242.8	883.3	339.9	470.8

We select three domain-specific lectures to collect a variety of behaviors while watching each university-level lecture. Also, we collect participants from two different language domains to verify their robustness or versatility. We issued an MCQ and the survey using Google Forms [1]. The MCQ for each lecture is ten questions with four choices. A unique aspect of this question design is that each corresponds to 1, 5, 10, 15, 20, 25, 30, 35, 40, and 50 minutes of viewing time. For example, participants can answer the first question if they understand the beginning part of the video for a minute, and they can answer the second question if they watch for five minutes. This setting allowed us to discover whether participants understood the lecture in each time segment. A survey asked two questions: "What

[1] `https://www.google.com/intl/en/forms/about/`

TABLE 4.11: Comparison of participants' self *unconfident* annotations and question-solving result for participants in Japan.

Participant	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Lecture 4 (Music) - Q1	TP	TP	FP	TP	TN	TP	TP	TP	TP	TP
Lecture 4 (Music) - Q2	TN	TP	FP	FP	TP	TP	TP	TP	TP	TP
Lecture 4 (Music) - Q3	TP	TP	FP	TP	TN	TP	TP	TP	TP	FN
Lecture 4 (Music) - Q4	FP	TN	FP	TP	TP	TP	TP	FP	TP	TP
Lecture 4 (Music) - Q5	TN	TP	TP	TP	TP	TP	TP	FP	TP	TN
Lecture 4 (Music) - Q6	FP	FN	FP	TP	FN	TP	TP	FP	FP	FP
Lecture 4 (Music) - Q7	TN	TP	FP	TN	TP	TP	TN	FN	FP	TN
Lecture 4 (Music) - Q8	TP	FP	TP	TP	FP	TP	TP	FP	FP	FP
Lecture 4 (Music) - Q9	TN	TP	TP	TP	TP	TN	TP	TP	FP	FN
Lecture 4 (Music) - Q10	TN	TP	FP	TP	TP	TP	TP	FP	FP	TN
Min continuous *unconfident* time (s)	20.6	30.8	452.1	305.4	30.8	54.4	4.5	103.1	78.8	27.7
Max continuous *unconfident* time (s)	144.3	142.7	1023.3	305.4	142.7	100.5	24.9	535.4	313.0	321.2
Mean continuous *unconfident* time (s)	56.8	68.4	711.9	305.4	68.4	77.5	17.7	339.1	194.9	103.1
Total *unconfident* time (s)	341.1	478.8	2847.7	305.4	478.8	155.0	141.9	1356.4	1559.8	721.8
Lecture 5 (Physics) - Q1	TP	TP	FN	TP	TN	TN	TN	TP	TN	TN
Lecture 5 (Physics) - Q2	TN	TN	TN	TN	TN	TP	TP	TP	TN	TN
Lecture 5 (Physics) - Q3	TP	TP	TP	TP	TP	TP	TP	TP	FP	TP
Lecture 5 (Physics) - Q4	TP	TP	FP	TN	TP	TP	TN	TN	FN	TN
Lecture 5 (Physics) - Q5	FP	TP	FP	TP	TP	TP	TP	TP	FP	TP
Lecture 5 (Physics) - Q6	FN	TN	FP	TP	TN	FN	FN	TN	FN	TN
Lecture 5 (Physics) - Q7	TP	TP	TP	TP	TN	TP	TP	TP	TP	TP
Lecture 5 (Physics) - Q8	TP	TP	FP	TP	TP	TP	TP	TP	TP	TP
Lecture 5 (Physics) - Q9	TP	FP	FP	TP	FP	TP	TP	TP	FN	TP
Lecture 5 (Physics) - Q10	TN	TN	FP	TP	TN	TP	TN	FP	TP	TP
Min continuous *unconfident* time (s)	60.0	15.6	60.0	146.2	60.0	283.6	32.3	493.9	64.6	49.6
Max continuous *unconfident* time (s)	138.4	180.9	600.1	277.9	133.8	283.6	64.6	493.9	323.1	120.6
Mean continuous *unconfident* time (s)	103.8	82.4	284.6	191.7	101.5	283.6	46.1	493.9	232.8	94.0
Total *unconfident* time (s)	623.2	247.4	1708.0	575.3	609.3	283.6	276.9	493.9	1629.6	376.0
Lecture 6 (Computer Science) - Q1	TP	TP	FP	TP	TP	TP	TP	TP	TP	TP
Lecture 6 (Computer Science) - Q2	FN	TP	FP	TN	TP	TN	TP	TN	TN	TN
Lecture 6 (Computer Science) - Q3	TN	TP	FN	TN	TN	TP	TP	TP	TN	TN
Lecture 6 (Computer Science) - Q4	FP	TP	TP	TN	TN	TP	TP	TP	TN	TN
Lecture 6 (Computer Science) - Q5	TP	TN	TP	TP	TP	TP	TP	TN	TN	TP
Lecture 6 (Computer Science) - Q6	FP	TP	TN	TP	TN	TP	TP	TN	TN	FN
Lecture 6 (Computer Science) - Q7	TP	TN	FN	TN	TN	TN	TP	TN	TN	TN
Lecture 6 (Computer Science) - Q8	FP	TP	TP	TP	TP	TP	TP	TP	TP	TP
Lecture 6 (Computer Science) - Q9	TP	TP	FP	TN	TN	TP	TP	TP	TN	TN
Lecture 6 (Computer Science) - Q10	TN	TP	TN	TP	TP	TP	TP	TN	TP	TP
Min continuous *unconfident* time (s)	118.9	0.0	272.8	0.0	43.2	36.6	0.0	83.1	59.7	73.5
Max continuous *unconfident* time (s)	281.6	0.0	751.9	0.0	156.3	36.6	0.0	116.4	59.7	186.0
Mean continuous *unconfident* time (s)	220.1	0.0	441.4	0.0	80.4	36.6	0.0	96.4	59.7	119.7
Total *unconfident* time (s)	880.5	0.0	1324.2	0.0	482.4	36.6	0.0	289.4	59.7	359.1

did you find easy (difficult) to understand about the lectures?". This perspective supports understanding why the participant felt *unconfident* about the video lecture.

Data Acquisition Tool

For data collection, we use a remote eye-tracker (Tobii 4C with an academic license) to record precise eye movements on the lecture video. The eye-tracker collects timestamps, pupil diameters, and x and y locations of the gaze. We also used the Microsoft Surface Studio webcam to record facial and body information. The webcam was 30.00 frame-per-second, and the screen resolution was 1280×720 pixels. We also collect self-annotation of

TABLE 4.12: Amount of each TP, TN, FP, and FN comprehension state labels.

Location	TP (True Positive)	TN (True Negative)	FP (False Positive)	FN (False Negative)
Germany	187	52	24	37
Japan	168	74	42	16
Total	355	126	66	53

the labeling of *unconfident* using Label Studio [168]. The tool is an open-source graphical user interface application, making it easy for anyone to annotate.

4.4.2 Results and Discussion

This section will detail the dataset balance and confidence/comprehension estimation results.

Dataset Balance

In this section, we summarize the collected dataset. This section is designed to provide valuable support for future researchers contemplating the utilization of our publicly available datasets. Figure 4.21 shows an output of gaze data (right and left pupil diameter difference) and self-reported *confidence* state compared with time. The red-colored rectangle indicates a time duration when the participant self-reported as *unconfident* in the video lecture. As the sample shows, the time duration of the *unconfident* state differs between the example of *Participant 5* and *Participant 8*. Looking more in detail in the Table 4.10 and Table 4.11, each shows the results of the comparison between participants' self-reported *unconfident* annotations and question-solving results in Germany and Japan. In the table, each TP, TN, FP, and FN means as below:

- TP (True Positive): Participant annotated as *confident* and made a *correct* answer to the MCQ.
- TN (True Negative): Participant annotated as *confident* and made a *wrong* answer to the MCQ.
- FP (False Positive): Participant annotated as *unconfident* and made a *correct* answer to the MCQ.
- FN (False Negative): Participant annotated as *unconfident* and made a *wrong* answer to the MCQ.

The correct answers are represented in the table using a gradient of green, and the incorrect answers are indicated in red. The intensity of the color reflects the participants' confidence level when providing the annotations while viewing the respective videos. Table 4.12 presents a comprehensive overview of the frequency counts for each of the four comprehension state labels (TP, TN, FP, and FN). The total count for the correct answers in the multiple-choice questions (MCQ) was 421 (TP + FP), while the count for the incorrect answers was 179 (TN + FN).

TABLE 4.13: LOPOCV results of binary confidence state using Gaussian Naive Bayes, Decision Tree and Random Forest.

Dataset Features	Model	Accuracy	Precision	Recall	F1-Score
Rawdata	Gaussian Naive Bayes	0.518	0.522	0.519	0.500
	Decision Tree	0.510	0.510	0.510	0.505
	Random Forest	0.519	0.519	0.519	0.515
Fixation and Saccades	Gaussian Naive Bayes	0.502	0.517	0.502	0.359
	Decision Tree	0.514	0.514	0.514	0.513
	Random Forest	0.517	0.517	0.517	0.517
Fixation, Saccades, and Pupil diameter	Gaussian Naive Bayes	0.509	0.515	0.509	0.457
	Decision Tree	0.524	0.524	0.524	0.524
	Random Forest	0.537	0.537	0.537	0.537

The results reveal an intriguing pattern, with TP having the highest count and, interestingly, FN having the lowest. Upon a detailed examination of each question, it became evident that at least one participant answered correctly with confidence (TP), suggesting the straightforward nature of these questions for at least one participant. However, further analysis indicated that specific questions, such as "Lecture 1 - Q9", "Lecture 3 - Q10", "Lecture 5 - Q6", and "Lecture 6 - Q7", posed significant challenges, as only one or two participants could answer them correctly with confidence. These findings underscore the importance of a meticulous review during the analysis phase, as specific questions might necessitate revision due to their inherent difficulty levels.

Furthermore, we identified distinct characteristics associated with the annotation labels. Specifically, we investigated the minimum, maximum, and total duration of *unconfident* time within the first 50 minutes of the video. Given that our study focused solely on MCQs that could be answered within the lecture video's initial 50 minutes, any data beyond this timeframe should have been included in the analysis. Remarkably, participant P5 exhibited the minimum total *unconfident* time during lecture 2, while participants P12, P14, and P17 in lecture six all recorded 0.0 seconds within a total timeframe of 3000 seconds (50 minutes). In contrast, the maximum total *unconfident* time among all participants was recorded by participant P8 in lecture 2, amounting to 2358.5 seconds within the 3000-second timeframe (50 minutes). A post-survey analysis indicated that the complexity of the terminology was cited as a prominent difficulty during the lecture, whereas the provision of comprehensible examples and diagrams was attributed to heightened participant confidence.

Confidence Estimation

In this section, we explain the result of self-reported confidence estimation from gaze data. Self-reported confidence is in the binary class of *True* and *False*. The NaN values extracted by the eye-tracker are filled into 0. This approach helps extract the participant's blinks or off-screen behavior, which is highlighted as significant in the mind-wondering prediction research [186]. Using the dataset, we apply several machine learning models against confidence estimation. Table 4.13 shows the results of leave-one-participant-out cross-validation (LOPOCV). We compared results by extracting several features.

TABLE 4.14: LOPOCV results of binary confidence state using Gaussian Naive Bayes, Decision Tree and Random Forest in English.

Dataset Features	Model	Accuracy	Precision	Recall	F1-Score
Rawdata	Gaussian Naive Bayes	0.491	0.491	0.491	0.490
	Decision Tree	0.524	0.524	0.524	0.524
	Random Forest	0.518	0.518	0.518	0.518
Fixation and Saccades	Gaussian Naive Bayes	0.505	0.505	0.505	0.497
	Decision Tree	0.521	0.521	0.521	0.520
	Random Forest	0.520	0.520	0.520	0.519
Fixation, Saccades, and Pupil diameter	Gaussian Naive Bayes	0.501	0.501	0.501	0.490
	Decision Tree	0.530	0.530	0.530	0.530
	Random Forest	0.543	0.543	0.543	0.543

TABLE 4.15: LOPOCV results of binary confidence state using Gaussian Naive Bayes, Decision Tree and Random Forest in Japanese.

Dataset Features	Model	Accuracy	Precision	Recall	F1-Score
Rawdata	Gaussian Naive Bayes	0.528	0.528	0.528	0.527
	Decision Tree	0.510	0.511	0.511	0.505
	Random Forest	0.529	0.530	0.529	0.526
Fixation and Saccades	Gaussian Naive Bayes	0.506	0.513	0.506	0.428
	Decision Tree	0.513	0.514	0.513	0.513
	Random Forest	0.522	0.522	0.522	0.522
Fixation, Saccades, and Pupil diameter	Gaussian Naive Bayes	0.513	0.515	0.513	0.501
	Decision Tree	0.514	0.515	0.515	0.515
	Random Forest	0.527	0.527	0.527	0.526

We first use raw data as input to see the baseline estimation result. The features are *gaze x*, *gaze y*, *left pupil*, *right pupil*, and *pupil diameter*. As a result, the F1-Score is 0.515 using the Random Forest classifier. Then, we convert raw data into fixations and saccades referring to the calculation. We define a fixation as a period of steady eye gazes at one point for at least 80 to 100 ms [24]. A saccade is a ballistic and rapid eye movement from one fixation to another [136]. The result of the F1-Score was 0.517, using the Random Forest classifier.

Lastly, we calculate the *min*, *max*, and *mean* pupil diameter while calculating fixation and saccades. This is to input data on pupil diameter while fixating. As a result, the model improved slightly, and Random Forest performed an F1-Score of 0.537. We also apply the same logic with each English and Japanese lecture viewer as shown in Table 4.14 and Table 4.15. The results show that the prediction performance with feature-extraction-based machine learning could have performed better.

We then apply deep learning techniques towards sequential eye gaze data. Figure 4.22 shows each participant's accuracy results. The mean F1-score was 0.530 and did not perform well for the sequential deep learning approach.

4.4.3 Comprehension Estimation

This section delves into the outcomes of our gaze data-driven comprehension estimation. Comprehension is categorized into four classes: *True Positive (TP)*, *True Negative (TN)*,

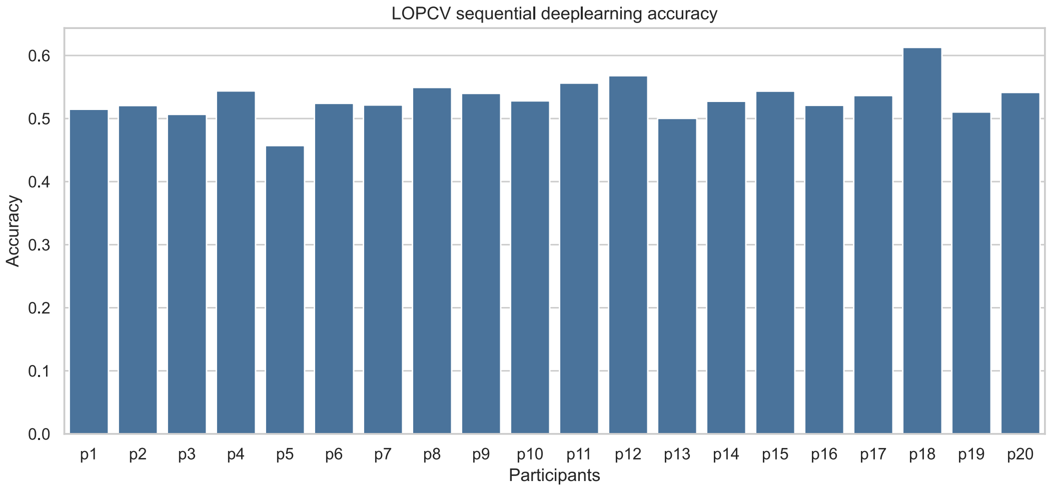


FIGURE 4.22: Leave-one-participant-out cross-validation results using sequential deep learning.

TABLE 4.16: LOPOCV results of 4 class comprehension state using Gaussian Naive Bayes, Decision Tree and Random Forest.

Dataset Features	Model	Accuracy	Precision	Recall	F1-Score
Rawdata	Gaussian Naive Bayes	0.251	0.243	0.251	0.197
	Decision Tree	0.254	0.254	0.254	0.251
	Random Forest	0.260	0.261	0.260	0.257
Fixation and Saccades	Gaussian Naive Bayes	0.253	0.257	0.253	0.237
	Decision Tree	0.256	0.255	0.256	0.255
	Random Forest	0.259	0.257	0.259	0.254
Fixation, Saccades, and Pupil diameter	Gaussian Naive Bayes	0.257	0.264	0.257	0.192
	Decision Tree	0.260	0.258	0.260	0.258
	Random Forest	0.266	0.263	0.266	0.259

TABLE 4.17: LOPOCV results of 4 class comprehension state using Gaussian Naive Bayes, Decision Tree and Random Forest in English.

Dataset Features	Model	Accuracy	Precision	Recall	F1-Score
Rawdata	Gaussian Naive Bayes	0.251	0.237	0.251	0.226
	Decision Tree	0.260	0.259	0.260	0.256
	Random Forest	0.262	0.260	0.262	0.256
Fixation and Saccades	Gaussian Naive Bayes	0.251	0.259	0.251	0.223
	Decision Tree	0.253	0.252	0.253	0.252
	Random Forest	0.260	0.258	0.260	0.255
Fixation, Saccades, and Pupil diameter	Gaussian Naive Bayes	0.257	0.272	0.257	0.226
	Decision Tree	0.255	0.253	0.255	0.252
	Random Forest	0.265	0.262	0.265	0.259

False Positive (FP), and *False Negative (FN)*. To handle NaN values extracted by the eye-tracker, we systematically fill them with zeros. This strategic approach aids in capturing participant blinks or off-screen behaviors, which have been identified as significant factors in mind-wandering prediction research [186].

TABLE 4.18: LOPOCV results of 4 class comprehension state using Gaussian Naive Bayes, Decision Tree and Random Forest in Japanese.

Dataset Features	Model	Accuracy	Precision	Recall	F1-Score
Rawdata	Gaussian Naive Bayes	0.256	0.249	0.256	0.207
	Decision Tree	0.265	0.266	0.265	0.260
	Random Forest	0.271	0.272	0.271	0.266
Fixation and Saccades	Gaussian Naive Bayes	0.262	0.260	0.262	0.230
	Decision Tree	0.255	0.255	0.255	0.255
	Random Forest	0.266	0.267	0.266	0.265
Fixation, Saccades, and Pupil diameter	Gaussian Naive Bayes	0.259	0.255	0.258	0.246
	Decision Tree	0.265	0.265	0.265	0.264
	Random Forest	0.278	0.279	0.278	0.276

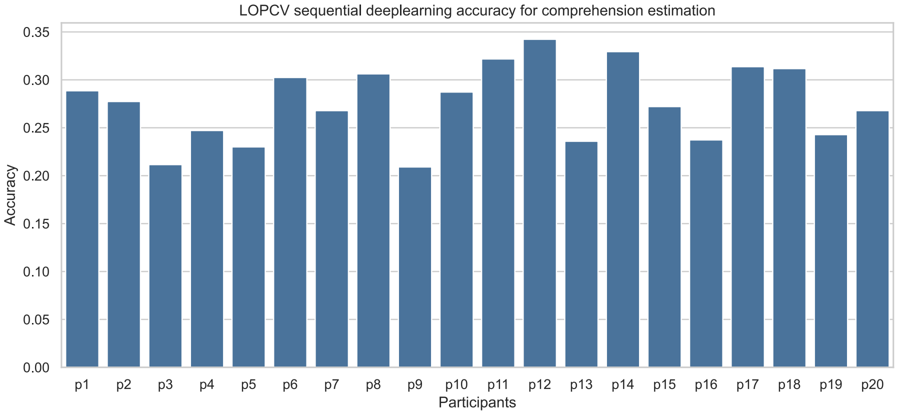


FIGURE 4.23: Leave-one-participant-out cross-validation results of estimating comprehension using sequential deep learning.

Employing the dataset, we leverage various machine learning models for confidence estimation, and the outcomes of leave-one-participant-out cross-validation (LOPOCV) are detailed in Table 4.16. Our comparative analysis involves the extraction of several features. Initially, we employ raw data as input to establish a baseline estimation result. Consequently, the Random Forest classifier yields an F1-Score of 0.257. Subsequently, we transform raw data into fixations and saccades based on our calculations, resulting in a slightly lower F1-Score of 0.254 using the Random Forest classifier. Finally, we incorporate the calculation of the min, max, and mean pupil diameter during fixation and saccades, providing data on pupil diameter during fixation. This refinement slightly improves, with the Random Forest classifier achieving an F1-Score of 0.259. Furthermore, we apply the same methodology to English and Japanese lecture viewers, as illustrated in Table 4.14 and Table 4.15. However, the results indicate that the prediction performance using feature-extraction-based machine learning is suboptimal for comprehension estimation.

We then apply deep learning techniques towards sequential eye gaze data. Figure 4.23 shows overall accuracy results for each participant. As a result, the mean F1-Score was 0.346, higher than all feature extraction methods proposed. However, the estimation of comprehension of video lecture viewers did not perform well for sequential deep learning methods.

4.4.4 Limitations and Future Work

One limitation of our experimental design is that we exclusively gathered data from three distinct lecture categories. While these categories were carefully chosen to represent diverse instructional styles, it is crucial to acknowledge that this limited selection may only partially encapsulate the broader spectrum of pedagogical approaches. Consequently, the findings and conclusions drawn from our study may not comprehensively reflect the entire landscape of lecture methodologies. Future research endeavors should consider expanding the scope to encompass a more extensive array of instructional modalities for a more robust understanding of the subject matter.

Another noteworthy limitation of this study is the depth of confidence in self-annotation. While we opted for binary classification in annotating confidence, it is crucial to acknowledge that confidence itself may not be inherently binary. Instead, it can manifest along a spectrum, encompassing varying degrees or levels of assurance. This multidimensional aspect implies that confidence holds a depth that extends beyond a simple dichotomy. Moreover, participants with elevated levels of self-esteem may exhibit a propensity to engage in more nuanced self-annotations regarding confidence, further contributing to the intricate and multifaceted nature of this psychological construct.

A third notable limitation is the relatively modest sample size of 20 participants, which may need to be increased to fully capture the range of individual differences and nuances within this study population. To increase the generalizability and statistical power of our findings, future studies should prioritize expanding the participant pool to encompass a more diverse and representative sample and to ensure a more comprehensive investigation of the phenomenon under investigation.

In our future research, we aim to enhance the robustness and depth of our investigations by expanding the participant cohort, thereby ensuring a more comprehensive exploration of the phenomena under scrutiny. Additionally, we plan to explore alternative machine learning approaches, such as implementing a sliding window methodology or incorporating advanced deep learning techniques like Long Short-Term Memory (LSTM) networks tailored to our dataset. The envisaged advancements also involve considering diverse input sources for estimation. For instance, we contemplate integrating presentation slides or the speaker's voice as additional input channels, thereby paving the way for a multi-modal sensing approach. This innovative strategy holds the potential to enrich our understanding and analysis, offering a more nuanced perspective on the intricate interplay between gaze data and comprehension estimation in diverse learning environments.

4.4.5 Conclusion

In this study, we collect eye-tracking data to predict confidence and comprehension levels during video lectures. A dataset encompassing 20 participants from Germany and Japan, where participants viewed a 50-minute lecture video spanning three domains, was gathered. Utilizing eye-tracking and webcam data, participants answered ten multiple-choice questions and employed an open-source annotation tool to mark segments where confidence waned.

We collect sensor data using the Tobii 4C eye tracker. Also, we collect multiple-choice question answers, survey results, and self-annotation of *unconfident* labels. We applied several machine-learning approaches and got an F1-Score of 0.346 for comprehension estimation. The prediction showed that eyes are difficult to predict the human comprehension level.

4.5 Discover Knowledge Sender: Presentation Skill Estimation

Presentation skills are important for understanding how knowledge is effectively encoded to deliver to a knowledge receiver. We also apply *TrackThink Camera* explained in Section 3.2 during preparation. Participants can search the website during a lecture to find resources supporting their ideas. Combining knowledge input and output skills, we aim to discover the characteristics of a skillful presenter.

4.5.1 Methodology

This study aims to estimate presentation skills from presenters' behavior. We collect both presenter and listener behaviors using cameras mounted on a tripod. The recorded presenter's videos are converted into facial and body key points using OpenFace and OpenPose. OpenFace is explained in Section 4.1.1.

Extracting Body Movements from Raw Video Frames Using OpenPose

OpenPose [26] is an open-source library for estimating the 2D pose of the whole body. The library jointly detects the human body, hand, facial, and foot of 135 key points on single images. Library users can choose and remove which key points to estimate. In this study, due to the camera angle and the table in front of the presenter, we decided to collect upper body movements, such as the right and left arms.

Understanding Presenter's Knowledge Input Skill

We collect presenters' preparation process using *TrackThinkCamera* explained in Section 3.2. The objective is to find how skillful presenter collect information to support their presentation interest and reliability. We collected web browsing features the same as Table 3.11.

Definition of skillful presenter

In this study, we classify skillful presenters by getting a presentation score from the audience. Audiences were asked to answer the score between 1 and 5 after each presentation. A skillful presenter is a presenter with a higher score than the average score ($\bar{X}$). Below is the calculation, where N stands for the number of participants and X_i is a score made by audiences.

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N} \tag{4.1}$$

4.5.2 Experimental Design

This study collected 15 university students (14 male and one female). All students are from Japan. Figure 4.24 shows the audience's condition in the classroom listening to the presentation. Figure 4.25 shows the condition of the presenter.

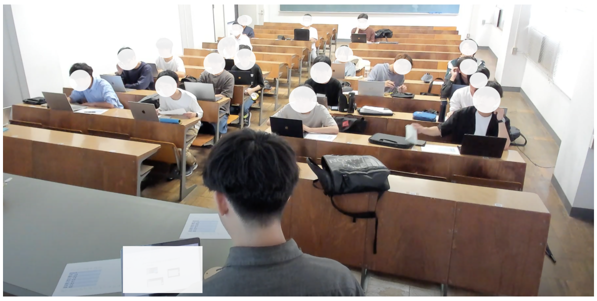


FIGURE 4.24: Camera recording of the class room while student conducting presentation.

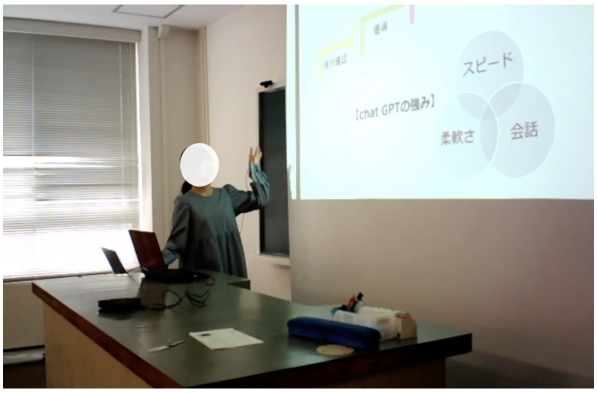


FIGURE 4.25: Camera recording of the presenter presenting in front of the class.

The presentation topic for each participant is “Future English Learning Using ChatGPT”. In order to make a presentation, participants used three lecture times, a total of 270 minutes, to search for information or evidence on the web. While searching for the evidence, we ask participants to use *TrackThinkCamera* explained in Section 3.2. Each presentation allows seven minutes to talk and three minutes for questions and answers. After the presentation, all students who were not with the presenter in the lecture evaluated the presenter with a score between 1 and 5.

4.5.3 Results and Discussion

Using OpenFace and OpenPose extracted features, we applied the Random Forest classifier. The result observed an F1-score of 0.815. The feature importance is shown in Figure 4.26. According to the result of feature importance, we found that *pose_Ty_min* is an important feature. This feature represents a minimum movement of the face. By looking at the raw video, we observed that low-scored presenter tend to look at their computer to follow the reading script. A high-scored presenter looks at both the audience and the slides. It implies that the presentation technique can be improved by giving feedback to the presenter on the head movements. A potential system could make the presenter change their behavior to realize and look at the audience frequently.

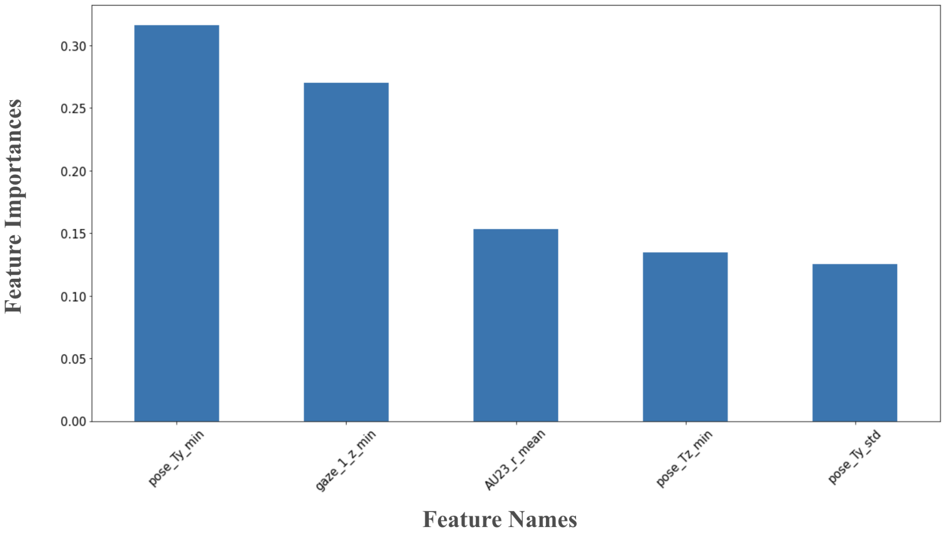


FIGURE 4.26: Camera recording of the presenter presenting in front of the class.

4.5.4 Limitation and Future Work

This study focuses on collecting presenters' and listeners' behaviors using cameras. However, we did not evaluate the listener's behaviors and only focused on the presenter. The study can still work on looking at, for example, speaker and listener synchronization. The collaborative work can be done with Osaka Metropolitan University to analyze the data further. In this study, we only focus on the presenter's behavior to understand the characteristics of skillful presenters.

Another future work is to implement an intervention system according to this finding. Since head motion or looking at the audience is important for presentation skills, the application can be implemented to support the presenter's realization. Our work mainly focuses on estimation, so in the future, we could implement a system to support behavior change smoothly to improve presentation skills.

4.5.5 Conclusion

We conduct a study of collecting presenters' and listeners' behavior using camera recordings. The study analyzes a presenter's face and body movements to estimate a skillful presenter. We collect 15 participants' presentation video recordings from the University in Japan. As a result, we gain an F1 score of 0.815 with a binary classified high and low-scored presenter. Looking at the importance of the feature, a vertical head motion was significant in identifying the skill.

Chapter 5

Intervention in Knowledge Transfer

This chapter explains the approach underlying knowledge transfer intervention applications. The primary objective of these interventions is to enhance the quality and effectiveness of human-to-human communication by fostering improved information exchange, mutual understanding, and collaboration in various contexts.

Section 5.1 explain *DiscussionJockey* [166] project, which leverages background music as a tool for enhancing online meetings. The application dynamically plays tailored background music for each participant in virtual meetings. The type of music varies based on individual participants' speech patterns and contributions. This intervention has demonstrated its ability to influence the flow of conversation, effectively modulating the amount of speaking time among multiple participants during discussions.

Section 5.2 explains *Metacognition-EnGauge* [171] project, which aims to estimate and visualize self-and-group engagement levels in real-time. The Metacognition-EnGauge system was developed to address these challenges by visualizing engagement levels in real-time through a gauge interface, using an *EnGauge* deep learning model. The system supports self-and-group metacognition augmentation and enhanced engagement in multiple participants' online communication.

Overall, this chapter shows the performance of the intervention of having a real-time actuation of background music and a real-time engagement gauge in online meetings. The study confirms controlling participants' utterances and increased engagement in communication.

5.1 DiscussionJockey: Smooth Interruption in Video Conference using BGM

Talking with others is essential to transfer knowledge and develop new ideas. Although COVID-19 has dramatically changed working styles, people value meetings as much or more than ever. Even after this pandemic ends, this trend will likely continue. However, compared to a face-to-face meeting, a video conference causes various problems, such as a lack of real-life communication with one's boss or colleague, video delay, and audio quality. Among those problems, we focused on the decrease in the quality of communication. For instance, understanding the other participants' feelings on recent video conference platforms has always been more challenging. It must be clarified if participants understand what a speaker is talking about, especially in online lessons. In addition, it is hard to notice even if others feel like saying something. This is because it lacks the information that comes from eye gaze, the direction of sound, or body language [145].

As typical trouble, a video conference makes it harder to interrupt the conversation, which can lead to a collision of conversation or deviation of engagement in the meeting. This comes from needing help judging the best time to interrupt the speaker. We propose a system that supports passing the batons of speech to others to solve such trouble. We have designed an online meeting bot that manipulates a speaker's speech rate using background music and makes space for a participant who wants to start talking. We chose background music as an intervention to give implicit feedback rather than direct suggestions. For example, Zoom has a raise-hand button, but some people are not willing to use that button to avoid interrupting the speaker.

Our research goal is to use a bot to do what people hesitate to do by themselves. When people interrupt other people's speech, we implicitly let the speaker know that others are trying to say something. This work is organized as follows. We first describe our survey findings, which support the idea that music can control speech rate. We then describe how the system we designed works. Next, we review the setting and result of the pilot study. Finally, we discuss our findings, including the problems we must solve.

5.1.1 Architecture

Figure 5.1 shows an overview of the proposed system. Our system is implemented as a web application composed of client and server sides connected via Web Socket. We assume that meeting participants use a video conference application like Zoom, Teams, or Google Meet. We leverage the necessary functions for video conferencing to the respective applications and perform speech amount acquisition and audio stimulation on the proposed system.

Measurement of the Amount of Speech

On the client side, we mainly used JavaScript, especially React, to develop the application. It first captures the utterance of a participant with the Web API getUserMedia method. We limit the scope to the frequency of human beings' utterances. The graphs on the left side of

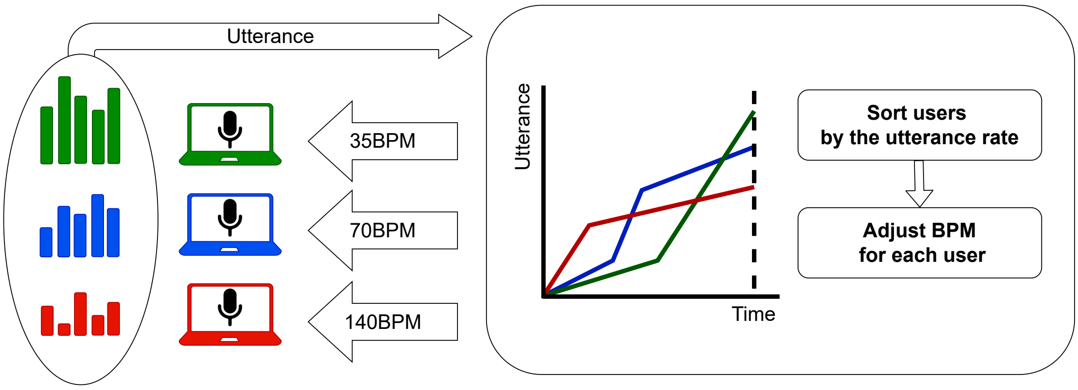


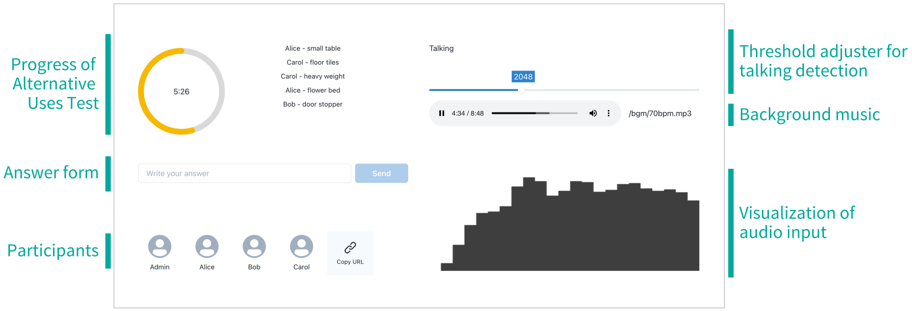
FIGURE 5.1: An overview of the workflow behind the proposed system

Figure 5.1 are examples of Fast Fourier Transform (FFT) descriptions. It is updated around 60 times per minute and sums up the whole value of every frequency in the range that we have limited. After that, if the score exceeds the threshold, it will be regarded as *Talking*. When judged as *Talking*, a variable is incremented and sent every five seconds. The threshold value depends on each environment, so participants can adjust it using the slider on the Web page. Before the experiment starts, they must find where it correctly says if they are *Talking* or *Not Talking*.

Intervention in Meeting

On the server side, it arranges the participants in a row according to the number of utterances as a total score since the beginning of the experiment, as Figure 5.1 shows. This ranking dynamically changes because of the consistent number of utterances sent from the client side. According to the ranking, the server sends commands to the client to adjust each user's beats per minute (BPM). If there are three participants, the person who speaks the most will hear the slowest background music (BGM): 35 beats per minute (BPM). The second engaged person will hear the medium-fast BGM, 70 BPM, and the last person will hear the fastest BGM, 140 BPM. This BGM will also dynamically change when the ranking is updated.

We chose 140 BPM BGM as the intervention because we expected upbeat music to affect the least engaged participant positively. Hosseini *et al.* [72] have studied the type of music that fosters extraordinary creativity in groups. The result of their study showed that upbeat BGM leads to divergent ideas. They hypothesized that upbeat music may decrease judgmental behavior during creative group tasks and inspire participants to share divergent perspectives. We then tried some beats and decided to use 35 BPM BGM for the most engaged participant in contrast. We do not have a scientific background for this, but we explained what each BPM BGM means before the experiment so users would know what each BGM implies for them. What kind of BGM is appropriate for the intervention remains controversial, so we will find the best BPM in future experiments.



Alice - small table
Carol - floor tiles
Carol - heavy weight
Alice - flower bed
Bob - door stopper

FIGURE 5.2: An application interface for the proposed system.

5.1.2 Experimental Design

We conducted a pilot study involving six volunteers. We collected the data with three participants and then tried again with different types of participants. In this research, we focus on the *utterance rate* and the results of the alternative uses task (AUT) [62]. We define *utterance rate* as each participant's speech percentage from the beginning of the experiment. In AUT, participants are asked to list as many alternative usages of a given object (e.g., tennis ball, spoon, and hanger) as possible. We utilized AUT to measure the creativity and activeness of a group meeting by counting the number of answers.

Figure 5.2 shows all components that are displayed for all participants. During an experiment, participants submitted their answers about the AUT to the server using a form. This data about utterance ranking, music intervention, and answers to the task was recorded constantly with the time stamp. In addition, an experiment conductor (admin) could access the other web page, which contains buttons to start and stop recording. When the admin presses the start button, the ten-minute timer shown in Figure 5.2 starts. CSV files about utterances, music, and answers are downloaded when the timer stops.

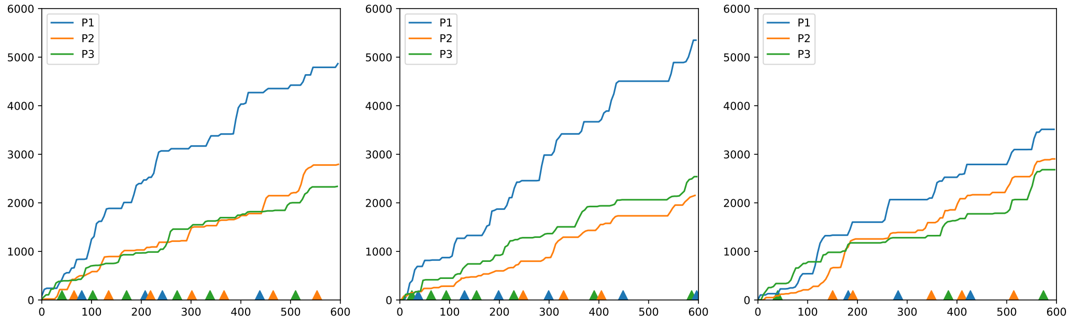
5.1.3 Results and Discussion

Figure 5.3 illustrates the transition of the amount of utterance and the timing of answers for each condition. The AUT objects were set to a tennis ball, spoon, and hanger.

Although P1 uttered more than the other participants in Conditions 1 and 2, it got closer to the rest in Condition 3. Table 5.1 summarizes their statistics. We calculated one participant's total utterances divided by the total utterances of all participants as *utterance rate*.

The *utterance rate* of each participant in Condition 3 was close to each other compared to the other two conditions. However, the total score of the number of answers was smaller than in the other conditions, so this is because the topic discussed in Condition 3 was more complex than the other questions.

Our pilot study has revealed some challenges. First, we need to consider the characteristics of the meeting participants. This research aims to determine the best time to cut into the conversation with a participant who is trying to speak by interrupting the speaker. Therefore, this system only works if other participants intend to speak. For instance, if one participant speaks only when he comes up with an answer and does not communicate with others, there



(A) Conition 1 - without BGM (B) Conition 2 - static BPM BGM (C) Conition 3 - dynamic BPM BGM

FIGURE 5.3: The cumulative amount of speech and timing of answers during 600 seconds

TABLE 5.1: Utterance rate and the number of answers of each condition

Condition	C1: without BGM			C2: static BPM BGM			C3: dynamic BPM BGM		
Participant	P1	P2	P3	P1	P2	P3	P1	P2	P3
Utterance rate [%]	48.7	27.9	23.4	53.3	21.4	25.3	38.6	31.9	29.5
The number of answers	4	7	6	6	3	7	3	5	3

is no meaning in making some space to start talking, and the utterance rate does not change. Another solution to this problem is to find different tasks that require more communication with others. When conducting a larger-scale experiment and evaluating the result statistically, we should also consider the difficulties of problems and the order of conditions.

There remain controversial issues, such as whether this intervention method was appropriate. We played the fastest BGM for the participant who spoke the least, but it is possible that it made the participant rush too much and talk less. Considering this, intervention might have to be the opposite of the original idea: fast music for the most vocal and slow music for the least vocal. Unlike background music, we can also use haptic feedback, like vibration, as an intervention. The advantage of audio feedback is that we do not have to prepare extra hardware devices. However, it sometimes takes too much attention and distracts participants more than needed. In contrast, mobility devices can intervene in the real world and interrupt the speaker more effectively than background music.

5.1.4 Conclusion

This paper introduced DiscussionJockey, a video conference bot that helps people join conversations by dynamically changing background music. Even if it is still a work in progress, our prototype and pilot study demonstrate the potential of audio stimuli during video conferences. Our underlying theme is leaving things people need help with to Artificial Intelligence. We hope to find an optimal way of intervention to realize natural conversation. In the proposed method, there is no function to identify who wants to say something and who speaks the most, and background music constantly interrupts the conversation. Therefore, in future work, we plan to integrate other feature modalities such as heart rate, facial expression, and eye gaze to identify who wants to start talking.

5.2 Metacognition-EnGauge: Understanding Self-and-Group Engagement Levels

Online meetings have expanded worldwide with the COVID-19 pandemic, opening the possibility of remotely collaborating. The opportunity opens up many people to work anytime and anywhere worldwide. However, online meetings often have difficulty communicating due to the lack of non-verbal behavior information [177]. Online meetings only provide upper body information and the need to understand behaviors like eye contact. With these issues in concern, supporting participants to help understand other meeting attendees' states is significant.

As explained in Section 4, we implement behavior and cognitive recognition systems like *DisCaas* and *EnGauge*. The *DisCaas* project aims to recognize non-verbal micro-behaviors such as nodding. The *EnGauge* project aims to recognize engagement levels only from webcam recordings, excluding voice information. Both systems run under online meeting conditions, which require only the upper body information. These projects' limitations were the need for more intervention or the application use case evaluation.

In this study, we demonstrate an application *Metacognition-EnGauge*. The system aims to augment the ability of the metacognition. We visualize the self and group engagement levels in real-time using a gauge interface. This work compares the effects of non-, self-, and group-metacognition augmentation interventions on visualizing engagement levels. The engagement level estimation uses *EnGauge* model proposed in Section 4.2. The deep-learning-based engagement levels prediction model [175]. To understand the effect of the application, we set three research questions for this project,

RQ1 Does self-metacognition augmentation of engagement levels support participants to be engaged in the meetings?

RQ2 Does group-metacognition augmentation of engagement levels support participants to be engaged in the meetings?

RQ3 Do participants prefer none, self, or group-metacognition augmentation intervention in the online meetings?

Research questions 1 and 2 aim to discover quantitative analysis of the self-and-group metacognition augmentation by examining whether the results increase participants' engagement levels during online meetings by applying the intervention. Research question 3 aims to discover qualitative analysis of participants by looking at the subjective feedback. This project contributes to understanding if metacognitive augmentation supports behavior changes.

5.2.1 Architecture

This section explains the model used for engagement level estimation and the architecture of how real-time conversion of engagement level into a percentage/gauge application works.

Engagement Levels Estimation Model

This study uses the engagement level detection model EnGauge [175]. The model uses data from 24 participants with high, middle, and low engagement levels. A role-playing approach collects each of the three different engagement levels. A high-level engagement participant focuses only on speaking, a middle-level engagement participant focuses on speaking and taking notes, and a low-level engagement participant ignores the conversation and reads a document. The prediction pipeline uses a facial cropped image into MobileNetV2 [146] based deep-learning model to classify the engagement.

Application Flow

The engagement level estimation model proposed in Section 5.2.1 can predict three-level engagement classified results. In order to make the result reliable in real-time and sequential, we applied time-concerned engagement level percentage prediction. Our application collects facial images from a webcam. We utilize HTML and JavaScript to capture a webcam frame image at five-second intervals during online meetings by recording the video screen-view. The captured webcam frame image then posts a request to the engagement detection model-mounted server. The server then predicts whether the requested participant webcam frame image is in a high, middle, or low engagement level state. This result is then stored in an array format as a result of engagement prediction in time-series (X_i). The engagement level estimation in percentage is done using the following formula:

$$Y(\%) = \frac{\sum_{i=1}^{N} X_i(\%)}{N} \tag{5.1}$$

Y represents the result of the engagement level in percentage. It will be a response from the server towards the client application. X_i is an engagement classification result, it is either 100% (high engagement), 50% (middle engagement), or 0% (low engagement). N is the number of elements. In this pilot application, we choose $N = 5$, which means the last five results of engagement classification inside X_i used to calculate Y. This approach supports gaining engagement-level prediction results into a time-sequential concern output. Once the client application receives a response from the server, it moves the gauge interface needle to indicate the value. The higher the engagement, the more the gauge needle moves towards 180 degrees of the gauge, and when it gets lower, it shifts to zero degrees. Therefore, the participants can check the prediction result every five seconds as feedback. We extracted the engagement level result into CSV to compare participant engagement levels in each experiment condition.

5.2.2 Data Collection

In this section, we explain the participants' background, the experimental condition, and details about the post-survey.

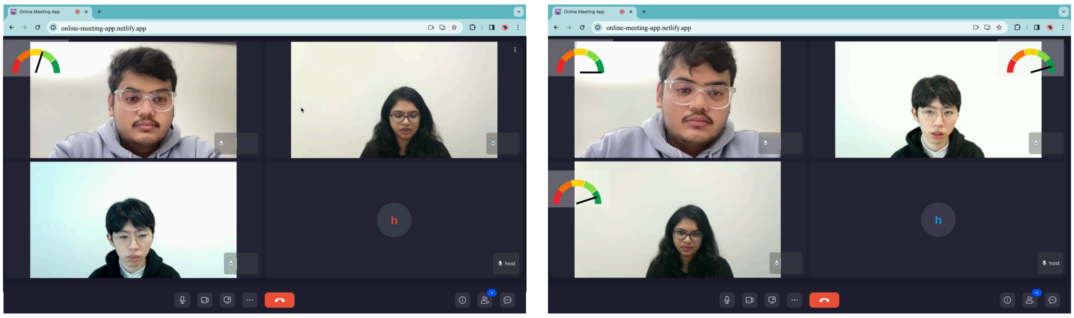

Online Meeting App
online-meeting-app.netlify.app
h
host
Online Meeting App
online-meeting-app.netlify.app
h
host

FIGURE 5.4: Engagement levels are presented as gauge in the online meeting. Self-and-group metacognition augmentation are shown.

Participants

We recruited 18 participants (Four females, 13 males, and one preferred not to say). Nationalities are ten Japanese, six Indian, one German, and one Chilean. Participants were between 23 and 31 years old, and the mean age was 26. They are either workers or university students in Japan and Germany. We obtained consent from all participants before the experiment. The General Data Protection Regulation (GDPR) is included for participants in Germany. All participants were allowed to opt out of the experiment at any time.

Experimental Design

In this study, we request participants to join from anywhere in remote conditions. In each experiment, three participants are asked to be in a group per session. In each session, we asked the group to discuss for ten minutes. We asked participants to join under three different online meeting conditions.

C-1 Online meeting without intervention.

C-2 Online meeting with self-metacognition augmentation by *EnGauge* intervention.

C-3 Online meeting with group-metacognition augmentation by *EnGauge* intervention.

Figure 5.4 explains the conditions of the second and third trials. The self-metacognition augmentation gives one gauge interface feedback to the participant. The group-metacognition augmentation gives all (three) participants' engagement gauge interface feedback to the participant. We asked participants to work on *would you rather questions* for the group discussion. Examples of topics are below,

T-A Would you rather live in a hot or cold country?

T-B Would you rather gain physical strength or brain intelligence augmentation?

T-C Would you rather go to the future or the past?

These topics were chosen because they are open questions for anyone to answer from their perspective. We were also required to ask participants to give supportive reasons for

TABLE 5.2: Application intervention conditions and topics for each experiment groups.

Condition and Topic	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Condition 1 (C-1)	T-A	T-A	T-B	T-B	T-C	T-C
Condition 2 (C-2)	T-B	T-C	T-A	T-C	T-A	T-B
Condition 3 (C-3)	T-C	T-B	T-C	T-A	T-B	T-A

their decision. Therefore, participants will continue to discuss until the session is over. Table 5.2 shows the topics for the experimental conditions presented to each group. Topics are presented differently to avoid differences in engagement based on topic interest.

Post Survey

Participants were asked to answer post-survey after each of the three experimental conditions. For post-survey, we used Google Forms [1] . In the post-survey, we asked the following questions,

Q1 What was the conclusion against the topic?

Q2 Please give the supportive reasons of the conclusion.

Q3 In your perspective, how engaging were you in the meeting? (Scale with 0 to 10)

Q4 How was the online meeting application? (Scale of 0 to 10)

Q5 Write any comments in general about the experiment.

Q6 How was the EnGauge system? Any difficulties? (Scale of 0 to 10)

Q7 Did you have any change in behaviors when using the application?

Q8 Can you give us opinions for the improvement of the application?

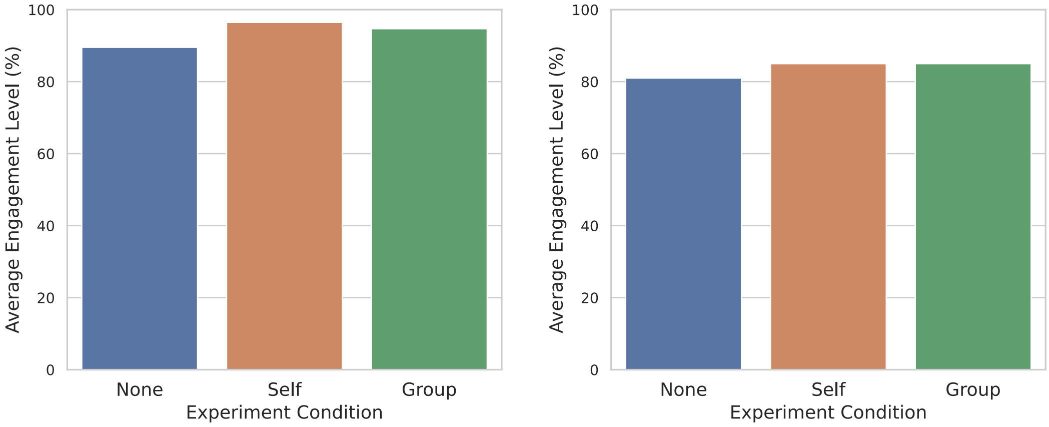
Q9 Which condition do you prefer? (Choose one from condition 1 to 3)

Regarding the questions, participants answered Q1-5 for all three conditions. For Q6-8, participants answer if they have experimented on self or group intervention. Lastly, answering Q9 was done once they finished the whole experiment.

5.2.3 Results and Discussion

This section states the experiment's results and discusses the answers to three research questions. Figure 5.5 shows the average engagement levels of participants in each condition, with deep-learning estimation and participant subjective feedback. Figure 5.6 shows the post-survey result of user preference on the intervention.

[1] https://www.google.com/intl/en/forms/about/



(A) Estimated by EnGauge deep-learning model. (B) Calculate from Likert scale in post-survey (Q3).

FIGURE 5.5: Comparison of the average engagement level of participants in each different intervention condition. Overall, the average engagement of all participants is highest for self-metacognition augmentation intervention. The participant preference is for group-metacognition augmentation intervention.

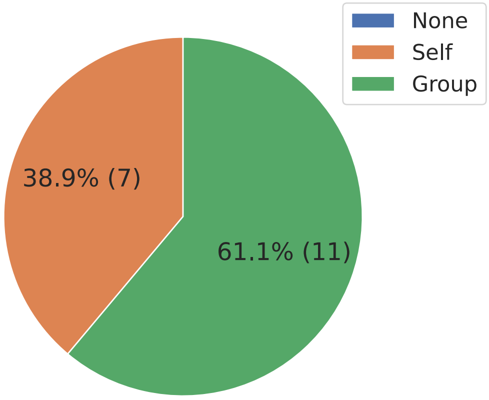


FIGURE 5.6: Participant preference of the self-and-group metacognition augmentation intervention.

RQ1: Does self-metacognition augmentation of engagement levels support the participants' engagement in the meetings?

The answer to RQ1, Figure 5.5 shows that self-metacognition augmentation performs an average engagement level during online meetings higher than without the intervention. Deep learning and subjective post-survey feedback were estimated to be higher than no intervention.

Looking at the post-survey, in Q7, participants mentioned some positive feedback. "*I tried hard to be more engaged*". "*I felt that I responded more actively and could express my opinions appropriately while still listening to others' opinions*". "*I was motivated to increase the gauge once the gauge went down*". "*I tried to speak when I saw the gauge and realized that I am not talking*". Also, we received some negative feedback. "*I think it was a little difficult to concentrate on both the discussion and the gauge*". "*Sometimes I moved my attention too much to the gauge*". "*No, since the gauge was not changing much*".

Concerning the positive feedback, the intervention can change behavior to encourage people to speak. Not only speaking but listening behavior is actively changed by nudging active response. Also, the gamification perspective has been reported as motivating the increase in the gauge meter. Against the negative feedback, we found that the user interface gives too much information for participants to focus on two things. Also, due to the experiment setup, we found that participants always tried actively to talk. Hence, the gauge did not move much.

RQ2: Does group-metacognition augmentation of engagement levels support participants to be engaged in the meetings?

The answer to RQ2, Figure 5.5 shows that group-metacognition augmentation performs an average engagement level during online meetings higher than without the intervention. Deep learning and subjective post-survey feedback were estimated to be higher than no intervention.

Looking at the post-survey, in Q7, participants mentioned some positive feedback. "*I checked other people's engagement carefully*". "*I paid attention to the utterance rate of other people*". "*I paid attention to who was most engaged at that time, and I tried to be engaged.*". "*When other people's gauge decrease, I tried to change how to talk and describe*". "*I am a person who actively behaves to show others that I am highly engaged. With the system, I became more careful to keep the gauge higher*". "*I carefully check when other people get low engagement*". Also, we received some negative feedback. "*No, because we were all engaged*". "*No it was like regular conversation*".

According to the positive feedback, group-metacognition augmentation performs significantly in behavior change. For example, participants carefully check whether they are speaking too much and other people less. Participants also actively tried to improve their ways of talking by observing each other's engagement. However, some participants did not think the intervention was powerful because they did not experience a change in behavior with the gauge.

RQ3: Do participants prefer none, self, or group-metacognition augmentation intervention in the online meetings?

In the answer to RQ3, we found that most participants prefer group-metacognition augmentation. Figure 5.6 shows the result in the pie-chart of the intervention preference. 11 participants voted for group-metacognition augmentation as their preference.

Looking at the result of the quantitative feedback in Figure 5.5, the average engagement is higher for the self-metacognition augmentation. The subjective feedback had the same average engagement for self and group. This means that subjective engagement can be improved by using group-metacognition augmentation. However, according to the quantitative results, self-metacognition augmentation should also be important.

Also, the participant gave feedback that "*I am a shy person, so it was easier for me to focus only on myself. I felt open-minded people would prefer a system like condition three that allows them to understand what is happening around them*". The comment supports our direction that the interface should be personalized according to the participant in the online meeting to augment or support their communication skills.

5.2.4 Conclusion

In this work, we verified the effectiveness of real-time augmentation of self-and-group engagement levels understanding using *Metacogniton-EnGauge*. Our study shows that self-and-group engagement levels feedback by gauge interface improve engagements throughout the meetings. The average engagement through the meeting performed best for self-metacognition augmentation. According to the survey, we found that most participants prefer group-metacognition augmentation.

Chapter 6

Conclusion

This chapter presents a summary of the research findings and discusses future work. Section 6.1 summarizes the contributions of this thesis by highlighting the main research problems and answers. Section 6.2 discusses the limitations of research in order to design future directions for the field.

6.1 Summary of the Thesis

Chapter 1 proposed three research questions: “How to discover the existing domain knowledge?", “How to visualize the state of knowledge transfer?”, and “How to accelerate the knowledge transfer?”. In order to answer three questions, we have worked on several projects. I have answered the questions with experiments summarized in Table 6.1.

TABLE 6.1: Domain knowledge estimation tools, activity and social cognitive state recognition, and intervention projects.

Section	Project	Input	Output	Publication
3.1	TrackThinkTS	User browsing actions	User action, tab related, contextual	[107]
3.2	TrackThink Camera	Browsing logs, webcam	User action, tab related, contextual, facial recording	[176]
3.3	Appearance Based Eye Tracking	Webcam	Gaze prediction	[17]
3.4	TrackThink Dashboard	Browsing logs, webcam	Flow-chart, pie-chart	[173]
4.1	DisCaaS	Camera	Micro-behaviors recognition	[33, 177]
4.2	EnGauge	Camera	Engagement levels recognition	[175]
4.3	Concern Gender as a Feature	Camera	Emotion recognition	[164]
5.1	DiscussionJockey	Microphone, Utterance	Background music	[166]
5.2	Metacognition-EnGauge	Camera	Engagement gauge	[171]

6.1.1 How to estimate the domain knowledge?

Chapter 3 tackles estimating domain knowledge using sensing technologies. We implement several logging applications and machine-learning models for this purpose.

Section 3.1 proposed our web browsing logger *TrackThinkTS*. The project focuses on collecting web browsing behaviors, tab management, and window operations. The application works on the Google Chrome extension. In combination with the programming logger *C2Room*, we collected coding and searching data from 33 participants. Among these participants, 13 have domain knowledge from university classes, and others still need to. Using the logged data, we estimate the domain knowledge by a mean accuracy of 0.95 with the Random Forest. We also extended the research to determine if we can extract useful knowledge from domain experts using force-directed and Sankey diagrams. As a result, the visualization approach helps us understand the helpful website more easily.

Section 3.2 proposed *TrackThinkCamera* integration of *TrackThinkTS* to further understand browsing behavior. To do so, we add synchronous webcam recording while browsing. This integration supports collecting facial recordings in parallel with the web browsing logs.

Section 3.3 proposed webcam-based eye-tracking in order to understand where participants were looking in the browser while searching. We collected 17 participants in total for the data collection. After applying several deep-learning models, VGG16 performed well on estimating 3.375 ± 0.891 cm pixel coordinate. The prediction model is in the process of improving and integrating to *TrackThinkCamera*.

Section 3.4 proposed the dashboard for visualizing flow-chart and pie-chart of the web browsing and coding logs collected by *TrackThinkTS* and *C2Room*. We found personality characteristics in the problem-solving patterns by visualizing the search path into time sequential concern using a flow chart and quantitatively using a pie chart. The application supports a deeper subjective understanding for educators to know how humans think in programming. Further, the application can seek how domain experts dig the information while problem-solving in detail.

6.1.2 How to visualize the state of knowledge transfer?

Chapter 4 explains the recognition of the activities related to knowledge transfer. We implement several machine-learning models for the prediction.

Section 4.1 proposed the estimation model of micro-behaviors. We collect both online and offline meeting camera recordings of a total of 34 unique participants. Using the collected video recordings, we focused on nodding and speaking detection. When comparing offline and online meetings, we detected nodding and speaking with an average F1 Score of 0.60 and 0.66, respectively. After applying the deep-learning methods, we detected nodding and speaking with an average F1-score of 0.81 and 0.94 [33].

Section 4.2 proposed the estimation model of engagement in online meetings. We collected 24 participants' high, middle, and low engagement through a role-acting approach.

Using the collected data, we implement the sliding window approach and achieve the classification of three class engagement levels by an average F1 score of 0.467 for the leave-one-participant-out approach. We then applied several deep-learning models and achieved an average F1 score of 0.923 for the leave-one-participant-out approach.

Section 4.3 proposed an approach to considering gender in emotion detection. The study uses the open-source dataset DAiSEE to detect four emotions. Our approach showed that gender-specific data significantly enhances emotion recognition accuracy, particularly for emotions like engagement and frustration. When considering the imbalance in the collected human dataset, gender can be one of the preprocessing features for improving model prediction.

Section 4.4 worked on estimating the comprehension level of the participants while watching online video lectures. We collected 20 participants, ten of whom were from Japan and Germany. Applying the machine-learning approach, we got an F1-Score of 0.346 for comprehension estimation. The study did not confirm the estimation of comprehension level from gaze data. Section 6.2, we will discuss some limitations and future work about this study.

Section 4.5 proposed the estimation model of presentation skill estimation in the on-site/class presentation. We collected seven minutes of video presentation recordings from 15 university students from Japan. After applying OpenFace and OpenPose to extract the facial and body key points, we applied Random Forest to estimate presentation skill by an average F1-score of 0.815. The study shows that the estimation of skilled presenters can be estimated.

6.1.3 How to accelerate the knowledge transfer?

Chapter 5 shows works on intervention applications for knowledge transfer. We present two intervention systems and the result of a user study.

DiscussionJockey presented in Section 5.1. The application aims to control the quantity of speech balance between three people in online meetings. The study uses BPM (Beat Per Minute) and BGM (Background Music) to enhance or reduce the participants speaking. As a result, the dynamic BPM BGM controlled the amount of speech among participants.

Metacognition-EnGauge presented in Section 5.2. The application aims to see whether the metacognition-augmentation of self and group engagement levels will change the behavior. From the practice, we gain that both self and group metacognition of engagement enhance participants' overall engagement levels in online meetings. A more interesting finding is that the participant mentioned the behavior change of making an effort to engage others while using group metacognition augmentation. For that, the participant mentioned improving the ways of explaining so that listeners can understand the statement and not become low-engaged. Hence, our proposed application showed some proof of effectively engaging participants by changing their communication behaviors.

6.2 Limitations and Future Work

The presented work has some limitations that can be improved in the future. Here, we present several possibilities for the direction of this thesis.

Prediction model performance. In Chapter 4, we present prediction and classification models of the activity, engagement levels, emotion, comprehension levels, and presentation skill recognition. Among these five main prediction models, we did not achieve a high classification rate for video lecture comprehension level estimation from eye tracking. To improve the study, we should conduct a pilot study carefully to analyze the data with a small number of participants. This study proceeded without analyzing the collected data, and hence, we can improve in the future by considering analyzing facial recordings from the webcam.

Ground truth annotation process. The comprehension level estimation prediction result did not work well due to the data collection procedure. In the future, ways of annotation can be improved. In our study, we asked participants to keep watching 50 minute video. Instead of this approach, we could prepare multiple shorter lecture videos and make the comprehension level annotation precise. The annotation procedure can be improved by changing the experimental setup.

Variation of the dataset. In Section 4.2, *EnGauge* project presents a new approach to collecting the three engagement levels in online meetings with a role-acting approach. The study allows for an efficient collection of engagement levels. Same for Section 4.1, *DisCaas* project collect the meeting data with a 360-degree camera for offline meetings. The approach allows multiple participants to be collected sitting near the table. However, we experimented with the set condition in both projects and did not collect data entirely in the wild. For improvement in the future, we can apply the implemented prediction model in the wild for validation. Also, open-source, publicly available datasets can be used for model improvements.

Evaluation of the intervention application. In Chapter 5, we present an intervention application for enhancing behavior change in online meetings. *DiscussionJockey* and *Metacognition-EnGauge* confirmed the effectiveness of the intervention with a group in the online meetings. However, the proposed intervention system was studied with a defined number of participants. The application showed effectiveness in a small group, but it can be evaluated in a classroom scenario with a bigger group of participants in the future.

In this thesis, I proposed works on accelerating knowledge transfer by sensing and actuating social-cognitive states. TrackThinkTS, TrackThinkCamera, TrackThinkDashboard, DisCaaS, EnGauge, DiscussionJockey, and Metacognition-EnGauge were demonstrated. The application's goal is to understand human domain knowledge, activity and cognitive states, and intervention on knowledge transfer. This thesis proposes several approaches to understanding and accelerating knowledge transfer with technology. The subsequent interest is to improve the intervention technology to enable it to be used in real-life scenarios.

Bibliography

[1] K. Adachi, P. Lago, T. Okita, and S. Inoue, “Improvement of human action recognition using 3d pose estimation,” *Activity and Behavior Computing*, pp. 21–37, 2021.

[2] S. Adams and C. Hanson, “Mit scheme user's manual,” *Massachusetts Institute of Technology*, 1995.

[3] S. Adams and C. Hanson, “Mit scheme user’ s manual,” *Massachusetts Institute of Technology*, 1995.

[4] B. Adriel Aseniero, M. Constantinides, S. Joglekar, K. Zhou, and D. Quercia, “Meetcues: Supporting online meetings experience,” in *2020 IEEE Visualization Conference (VIS)*, 2020, pp. 236–240. DOI: 10.1109/VIS47514.2020.00054.

[5] A. A. Ahmed and M. S. Goodwin, “Automated detection of facial expressions during computer-assisted instruction in individuals on the autism spectrum,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’17, Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 6050–6055, ISBN: 9781450346559. DOI: 10.1145/3025453.3025472. [Online]. Available: https://doi.org/10.1145/3025453.3025472.

[6] K. Ahuja *et al.*, “Edusense: Practical classroom sensing at scale,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 3, 2019, Visited on May 19, 2023. DOI: 10.1145/3351229. [Online]. Available: https://doi.org/10.1145/3351229.

[7] J. A. Allen, S. Rogelberg, and J. C. Scott, “Mind your meetings: Improve your organization’ s effectiveness one meeting at a time,” *Quality Progress*, vol. 41, pp. 48–53, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:110492431.

[8] E. Andonova and H. A. Taylor, *Cognitive Processing*, vol. 13, no. 1, p. 7982, 20122012/08/01, ISSN: 1612-4790. DOI: 10.1007/s10339-012-0472-x. [Online]. Available: https://doi.org/10.1007/s10339-012-0472-x.

[9] A. Aula, R. M. Khan, and Z. Guan, “How does search behavior change as search becomes more difficult?” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’10, Atlanta, Georgia, USA: Association for Computing Machinery, 2010, 35–44, ISBN: 9781605589299. DOI: 10.1145/1753326.1753333. [Online]. Available: https://doi.org/10.1145/1753326.1753333.

[10] E. Babaei, N. Srivastava, J. Newn, Q. Zhou, T. Dingler, and E. Velloso, “Faces of focus: A study on the facial cues of attentional states,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’20, Visited on May 19, 2023, Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–13, ISBN: 9781450367080. DOI: 10.1145/3313831.3376566. [Online]. Available: https://doi.org/10.1145/3313831.3376566.

[11] J. Bae, V. Setlur, and B. Watson, “Graphtiles: A visual interface supporting browsing and imprecise mobile search,” in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’15, Copenhagen, Denmark: Association for Computing Machinery, 2015, pp. 63–70, ISBN: 9781450336529. DOI: 10.1145/2785830.2785872. [Online]. Available: https://doi.org/10.1145/2785830.2785872.

[12] E. Bailey and D. Kelly, “Developing a measure of search expertise,” in *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, ser. CHIIR ’16, Carrboro, North Carolina, USA: Association for Computing Machinery, 2016, 237–240, ISBN: 9781450337519. DOI: 10.1145/2854946.2854983. [Online]. Available: https://doi.org/10.1145/2854946.2854983.

[13] T. Baltrušaitis, P. Robinson, and L.-P. Morency, “Openface: An open source facial behavior analysis toolkit,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–10. DOI: 10.1109/WACV.2016.7477553.

[14] S. Bardzell and J. Bardzell, “Towards a feminist hci methodology: Social science, feminism, and hci,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’11, Vancouver, BC, Canada: Association for Computing Machinery, 2011, 675–684, ISBN: 9781450302289. DOI: 10.1145/1978942.1979041. [Online]. Available: https://doi.org/10.1145/1978942.1979041.

[15] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.

[16] H. Bello, B. Zhou, S. Suh, and P. Lukowicz, “Mocapaci: Posture and gesture detection in loose garments using textile cables as capacitive antennas,” in *Proceedings of the 2021 ACM International Symposium on Wearable Computers*, ser. ISWC ’21, Virtual, USA: Association for Computing Machinery, 2021, 78–83, ISBN: 9781450384629. DOI: 10.1145/3460421.3480418. [Online]. Available: https://doi.org/10.1145/3460421.3480418.

[17] A. Bhatt, K. Watanabe, A. Dengel, and S. Ishimaru, “Appearance-based gaze estimation with deep neural networks: From data collection to evaluation,” *International Journal of Activity and Behavior Computing*, vol. 2024, no. 1, pp. 1–15, 2024. DOI: 10.60401/ijabc.9.

[18] L. S. Bohannon, A. M. Herbert, J. B. Pelz, and E. M. Rantanen, “Eye contact and video-mediated communication: A review,” *Displays*, vol. 34, no. 2, pp. 177–185, 2013.

[19] G. Bradski, "The opencv library.," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[20] J. Bridges, "Gendering metapragmatics in online discourse: "mansplaining man gonna mansplain···" ," *Discourse, Context and Media*, vol. 20, pp. 94–102, 2017. [Online]. Available: `https://api.semanticscholar.org/CorpusID:149228968`.

[21] I. Brishtel, A. A. Khan, T. Schmidt, T. Dingler, S. Ishimaru, and A. Dengel, "Mind wandering in a multimodal reading setting: Behavior analysis & automatic detection using eye-tracking and an eda sensor," *Sensors*, vol. 20, no. 9, p. 2546, 2020.

[22] L. R. Brody, "Gender differences in emotional development: A review of theories and research," *Journal of Personality*, vol. 53, no. 2, pp. 102–149, 1985. DOI: `10.1111/j.1467-6494.1985.tb00361.x`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6494.1985.tb00361.x`.

[23] D. M. Bunce, E. A. Flens, and K. Y. Neiles, "How long can students pay attention in class? a study of student attention decline using clickers," *Journal of Chemical Education*, vol. 87, no. 12, pp. 1438–1443, 2010.

[24] G. Buscher, E. Cutrell, and M. R. Morris, "What do you see when you're surfing? using eye tracking to predict salient regions of web pages," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09, Boston, MA, USA: Association for Computing Machinery, 2009, 21–30, ISBN: 9781605582467. DOI: `10.1145/1518701.1518705`. [Online]. Available: `https://doi.org/10.1145/1518701.1518705`.

[25] E. Cai, R. Rossi, and C. Xiao, "Improving learning-based camera pose estimation for image-based augmented reality applications," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '23, Hamburg, Germany: Association for Computing Machinery, 2023, ISBN: 9781450394222. DOI: `10.1145/3544549.3585756`. [Online]. Available: `https://doi.org/10.1145/3544549.3585756`.

[26] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[27] J. Carletta *et al.*, "The ami meeting corpus: A pre-announcement," English, in *Machine Learning for Multimodal Interaction*, S. Renals and S. Bengio, Eds., Springer Berlin Heidelberg, 2006, pp. 28–39, ISBN: 978-3-540-32549-9. DOI: `10.1007/11677482_3`.

[28] M. Carrasco, E. Koh, and S. Malik, "Pophistory: Animated visualization of personal web browsing history," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '17, Denver, Colorado, USA: Association for Computing Machinery, 2017, 2429–2436, ISBN: 9781450346566. DOI: `10.1145/3027063.3053259`. [Online]. Available: `https://doi.org/10.1145/3027063.3053259`.

[29] J. Cen, T. Xu, and J. Yu, "Examining gender-oriented design features in computational toys and kits for young children," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23, Hamburg, Germany: Association for Computing Machinery, 2023, ISBN: 9781450394215. DOI: 10.1145/3544548.3581035. [Online]. Available: https://doi.org/10.1145/3544548.3581035.

[30] S. Centorrino, E. Djemaï, A. Hopfensitz, M. Milinski, and P. Seabright, "Honest signalling in trust interactions: Smiles rated as genuine induce trust and signal higher earnings opportunities," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:2315956.

[31] S. Chacon and B. Straub, *Pro git*. Springer Nature, 2014.

[32] T. M. Chaplin and A. Aldao, "Gender differences in emotion expression in children: A meta-analytic review.," *Psychological bulletin*, vol. 139, no. 4, p. 735, 2013.

[33] C. Chen, Y. Arakawa, K. Watanabe, and S. Ishimaru, "Quantitative evaluation system for online meetings based on multimodal microbehavior analysis," *Sensors and Materials*, vol. 34, no. 8, pp. 3017–3027, 2022. DOI: 10.18494/SAM3959.

[34] D. Cheng, P.-Y. Chi, T. Kwak, B. Hartmann, and P. Wright, "Body-tracking camera control for demonstration videos," in *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '13, Paris, France: Association for Computing Machinery, 2013, pp. 1185–1190, ISBN: 9781450319522. DOI: 10.1145/2468356.2468568. [Online]. Available: https://doi.org/10.1145/2468356.2468568.

[35] Y. Cheng, H. Wang, Y. Bao, and F. Lu, *Appearance-based gaze estimation with deep learning: A review and benchmark*, 2021. arXiv: 2104.12668 [cs.CV].

[36] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[37] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[38] S. Chu and J. Tanaka, "Interacting with a self-portrait camera using motion-based hand gestures," in *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, ser. APCHI '13, Bangalore, India: Association for Computing Machinery, 2013, pp. 93–101, ISBN: 9781450322539. DOI: 10.1145/2525194.2525206. [Online]. Available: https://doi.org/10.1145/2525194.2525206.

[39] J. F. Cohn and F. De la Torre, "Automated face analysis for affective computing.," 2015.

[40] J. Cryan, S. Tang, X. Zhang, M. Metzger, H. Zheng, and B. Y. Zhao, "Detecting gender stereotypes: Lexicon vs. supervised learning methods," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–11, ISBN: 9781450367080. DOI: 10.1145/3313831.3376488. [Online]. Available: https://doi.org/10.1145/3313831.3376488.

[41] H. Da Rocha, *Learn Chart. js: Create interactive visualizations for the web with chart. js 2*. Packt Publishing Ltd, 2019.

[42] B. De Carolis, F. D'Errico, N. Macchiarulo, and G. Palestra, " "engaged faces" : Measuring and monitoring student engagement from face and gaze behavior," in *IEEE/WIC/ACM International Conference on Web Intelligence - Companion Volume*, ser. WI '19 Companion, Visited on May 19, 2023, Thessaloniki, Greece: Association for Computing Machinery, 2019, 80–85, ISBN: 9781450369886. DOI: `10.1145/3358695.3361748`. [Online]. Available: `https://doi.org/10.1145/3358695.3361748`.

[43] M. Dehghani, G. Jagfeld, H. Azarbonyad, A. Olieman, J. Kamps, and M. Marx, "On search powered navigation," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, ser. ICTIR '17, Amsterdam, The Netherlands: Association for Computing Machinery, 2017, pp. 317–320, ISBN: 9781450344906. DOI: `10.1145/3121050.3121105`. [Online]. Available: `https://doi.org/10.1145/3121050.3121105`.

[44] S. Dhamija, "Learning based visual engagement and self-efficacy," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017, pp. 581–585. DOI: `10.1109/ACII.2017.8273659`.

[45] E. Di Lascio, S. Gashi, and S. Santini, "Unobtrusive assessment of students' emotional engagement during lectures using electrodermal activity sensors," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, 2018, Visited on May 19, 2023. DOI: `10.1145/3264913`. [Online]. Available: `https://doi.org/10.1145/3264913`.

[46] I. ud Din, S. Khusro, I. Ullah, and A. Rauf, "Semantic history: Ontology-based modeling of users' web browsing behaviors for improved web page revisitation," in *Proceedings of the Computational Methods in Systems and Software*, Springer, 2018, pp. 204–215.

[47] B. DiSalvo, D. Bandaru, Q. Wang, H. Li, and T. Plötz, "Reading the room: Automated, momentary assessment of student engagement in the classroom: Are we there yet?" *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 3, 2022, Visited on May 19, 2023. DOI: `10.1145/3550328`. [Online]. Available: `https://doi.org/10.1145/3550328`.

[48] G. J. Edwards, A. Lanitis, C. J. Taylor, and T. Cootes, "Statistical models of face images - improving specificity," *Image Vis. Comput.*, vol. 16, pp. 203–211, 1998. [Online]. Available: `https://api.semanticscholar.org/CorpusID:807316`.

[49] P. EKMAN and W. V. FRIESEN, "The repertoire of nonverbal behavior: Categories, origins, usage, and coding," *Semiotica*, vol. 1, no. 1, pp. 49–98, 1969. DOI: `doi:10.1515/semi.1969.1.1.49`. [Online]. Available: `https://doi.org/10.1515/semi.1969.1.1.49`.

[50] P. Ekman and W. V. Friesen, "Facial action coding system," *Environmental Psychology & Nonverbal Behavior*, 1978.

[51] T. Eloundou, S. Manning, P. Mishkin, and D. Rock, "Gpts are gpts: An early look at the labor market impact potential of large language models," *arXiv preprint arXiv:2303.10130*, 2023.

[52] S. Freeman *et al.*, "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the national academy of sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.

[53] L. Fridman, B. Reimer, B. Mehler, and W. T. Freeman, "Cognitive load estimation in the wild," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, Montreal QC, Canada: Association for Computing Machinery, 2018, 1–9, ISBN: 9781450356206. DOI: `10.1145/3173574.3174226`. [Online]. Available: `https://doi.org/10.1145/3173574.3174226`.

[54] Y. Fu *et al.*, "Robust subjective visual property prediction from crowdsourced pairwise labels," in *IEEE TPAMI*, 2016.

[55] K. A. Funes Mora, F. Monay, and J.-M. Odobez, "Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '14, Safety Harbor, Florida: Association for Computing Machinery, 2014, 255–258, ISBN: 9781450327510. DOI: `10.1145/2578153.2578190`. [Online]. Available: `https://doi.org/10.1145/2578153.2578190`.

[56] N. Gao, M. S. Rahaman, W. Shao, K. Ji, and F. D. Salim, "Individual and group-wise classroom seating experience: Effects on student engagement in different courses," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 3, 2022, Visited on May 19, 2023. DOI: `10.1145/3550335`. [Online]. Available: `https://doi.org/10.1145/3550335`.

[57] P. Garg, J. Santhosh, A. Dengel, and S. Ishimaru, "Stress detection by machine learning and wearable sensors," in *26th International Conference on Intelligent User Interfaces-Companion*, 2021, pp. 43–45.

[58] S. Gashi, E. Di Lascio, and S. Santini, "Using unobtrusive wearable sensors to measure the physiological synchrony between presenters and audience members," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, 2019, Visited on May 19, 2023. DOI: `10.1145/3314400`. [Online]. Available: `https://doi.org/10.1145/3314400`.

[59] X. Geng, Z.-H. Zhou, and K. Smithąles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2234–2240, 2007. [Online]. Available: `https://api.semanticscholar.org/CorpusID:8346560`.

[60] J. Grimmer, L. Simon, and J. Ehlers, "The cognitive eye: Indexing oculomotor functions for mental workload assessment in cognition-aware systems," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '21, Yokohama, Japan: Association for Computing Machinery, 2021, ISBN: 9781450380959. DOI: 10.1145/3411763.3451662. [Online]. Available: https://doi.org/10.1145/3411763.3451662.

[61] Z. Gu, F. Meng, and M. Farrukh, "Mapping the research on knowledge transfer: A scientometrics approach," *IEEE Access*, vol. 9, pp. 34 647–34 659, 2021.

[62] J. P. Guilford, "Creativity: Yesterday, today and tomorrow," *The Journal of Creative Behavior*, vol. 1, no. 1, pp. 3–14, 1967.

[63] A. Gupta, A. D'Cunha, K. Awasthi, and V. Balasubramanian, "Daisee: Towards user engagement recognition in the wild," *arXiv preprint arXiv:1609.01885*, 2016.

[64] S. Gupta, P. Kumar, and R. K. Tekchandani, "Facial emotion recognition based real-time learner engagement detection system in online learning context using deep learning models," *Multimedia Tools and Applications*, pp. 1–30, 2022.

[65] L. Haliburton, S. Y. Schött, L. Hirsch, R. Welsch, and A. Schmidt, "Feeling the temperature of the room: Unobtrusive thermal display of engagement during group communication," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 1, 2023. DOI: 10.1145/3580820. [Online]. Available: https://doi.org/10.1145/3580820.

[66] F. Hamidi, M. K. Scheuerman, and S. M. Branham, "Gender recognition or gender reductionism? the social implications of embedded gender recognition systems," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, Montreal QC, Canada: Association for Computing Machinery, 2018, 1–13, ISBN: 9781450356206. DOI: 10.1145/3173574.3173582. [Online]. Available: https://doi.org/10.1145/3173574.3173582.

[67] B. Hebda and T. Kryjak, "A compact deep convolutional neural network architecture for video based age and gender estimation," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2016, pp. 787–790.

[68] J. Hernandez, I. Riobo, A. Rozga, G. D. Abowd, and R. W. Picard, "Using electrodermal activity to recognize ease of engagement in children during social interactions," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14, Visited on May 19, 2023, Seattle, Washington: Association for Computing Machinery, 2014, 307–317, ISBN: 9781450329682. DOI: 10.1145/2632048.2636065. [Online]. Available: https://doi.org/10.1145/2632048.2636065.

[69] C. Hölscher and G. Strube, "Web search behavior of internet experts and newbies," *Computer networks*, vol. 33, no. 1-6, pp. 337–346, 2000.

[70] K. Holstein, G. Hong, M. Tegene, B. M. McLaren, and V. Aleven, "The classroom as a dashboard: Co-designing wearable cognitive augmentation for k-12 teachers," in *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, ser. LAK '18, Sydney, New South Wales, Australia: Association for Computing Machinery, 2018, 79–88, ISBN: 9781450364003. DOI: 10.1145/3170358.3170377. [Online]. Available: https://doi.org/10.1145/3170358.3170377.

[71] A. Hora, "Googling for software development: What developers search for and what they find," in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 2021, pp. 317–328. DOI: 10.1109/MSR52588.2021.00044.

[72] S. Hosseini, X. Deng, Y. Miyake, and T. Nozawa, "Head movement synchrony and idea generation interference–investigating background music effects on group creativity," *Frontiers in Psychology*, vol. 10, p. 2577, 2019.

[73] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017, Visited on May 19, 2023. arXiv: 1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861.

[74] I. Hsieh-Yee, "Research on web search behavior," *Library & Information Science Research*, vol. 23, no. 2, pp. 167–185, 2001.

[75] I. Huerta, C. Fernández, C. Segura, J. Hernando, and A. Prati, "A deep analysis on age estimation," *Pattern Recognition Letters*, vol. 68, pp. 239–249, 2015, Special Issue on "Soft Biometrics" , ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2015.06.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865515001683.

[76] S. Huynh, S. Kim, J. Ko, R. K. Balan, and Y. Lee, "Engagemon: Multi-modal engagement sensing for mobile games," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, 2018, Visited on May 19, 2023. DOI: 10.1145/3191745. [Online]. Available: https://doi.org/10.1145/3191745.

[77] dTosh Inc., *Product | c2room*, 2020. [Online]. Available: https://c2room.jp/.

[78] S. Ishimaru, S. S. Bukhari, C. Heisel, J. Kuhn, and A. Dengel, "Towards an intelligent textbook: Eye gaze based attention extraction on materials for learning and instruction in physics," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp '16, Heidelberg, Germany: Association for Computing Machinery, 2016, 1041–1045, ISBN: 9781450344623. DOI: 10.1145/2968219.2968566. [Online]. Available: https://doi.org/10.1145/2968219.2968566.

[79] S. Ishimaru and A. Dengel, "Arfled: Ability recognition framework for learning and education," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ser. UbiComp '17, Maui, Hawaii: Association for Computing Machinery, 2017, 339–343, ISBN: 9781450351904. DOI: 10.1145/

3123024.3123200. [Online]. Available: https://doi.org/10.1145/3123024.3123200.

[80] S. Ishimaru, T. Dingler, K. Kunze, K. Kise, and A. Dengel, “Reading interventions: Tracking reading state and designing interventions,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp ’16, Heidelberg, Germany: Association for Computing Machinery, 2016, 1759–1764, ISBN: 9781450344623. DOI: 10.1145/2968219.2968271. [Online]. Available: https://doi.org/10.1145/2968219.2968271.

[81] S. Ishimaru, K. Kunze, K. Kise, and A. Dengel, “The wordometer 2.0: Estimating the number of words you read in real life using commercial eog glasses,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp ’16, Heidelberg, Germany: Association for Computing Machinery, 2016, 293–296, ISBN: 9781450344623. DOI: 10.1145/2968219.2971398. [Online]. Available: https://doi.org/10.1145/2968219.2971398.

[82] S. Ishimaru, T. Maruichi, K. Kise, and A. Dengel, “Gaze-based self-confidence estimation on multiple-choice questions and its feedback,” in *Proceedings of the 2020 Symposium on Emerging Research from Asia and on Asian Contexts and Cultures*, 2020, pp. 8–8.

[83] S. Ishimaru *et al.*, “Cognitive state measurement on learning materials by utilizing eye tracker and thermal camera,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 08, 2017, pp. 32–36. DOI: 10.1109/ICDAR.2017.378.

[84] S. Ishimaru *et al.*, “Hypermind builder: Pervasive user interface to create intelligent interactive documents,” in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp ’18, Singapore, Singapore: Association for Computing Machinery, 2018, 357–360, ISBN: 9781450359665. DOI: 10.1145/3267305.3267667. [Online]. Available: https://doi.org/10.1145/3267305.3267667.

[85] A. L. Janin *et al.*, “The icsi meeting corpus,” *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03).*, vol. 1, pp. I–I, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:18614936.

[86] P. Jiang, F. Sun, and H. Xia, “Log-it: Supporting programming with interactive, contextual, structured, and visual logs,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’23, Hamburg, Germany: Association for Computing Machinery, 2023, ISBN: 9781450394215. DOI: 10.1145/3544548.3581403. [Online]. Available: https://doi.org/10.1145/3544548.3581403.

[87] J. C. Karremans and P. A. Van Lange, “Forgiveness in personal relationships: Its malleability and powerful consequences,” *European Review of Social Psychology*, vol. 19, no. 1, pp. 202–241, 2008.

[88] S. Kauffeld and N. Lehmann-Willenbrock, “Meetings matter: Effects of team meetings on team and organizational success,” *Small group research*, vol. 43, no. 2, pp. 130–158, 2012.

[89] H. Kaur, D. McDuff, A. C. Williams, J. Teevan, and S. T. Iqbal, “ “i didn’ t know i looked angry” : Characterizing observed emotion and reported affect at work,” in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’22, New Orleans, LA, USA: Association for Computing Machinery, 2022, ISBN: 9781450391573. DOI: 10 . 1145 / 3491102 . 3517453. [Online]. Available: https://doi.org/10.1145/3491102.3517453.

[90] R. Kawamura, S. Shirai, M. Aizadeh, N. Takemura, and H. Nagahara, “Estimation of wakefulness in video-based lectures based on multimodal data fusion,” in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, ser. UbiComp-ISWC ’20, Virtual Event, Mexico: Association for Computing Machinery, 2020, 50–53, ISBN: 9781450380768. DOI: 10.1145/3410530.3414386. [Online]. Available: https://doi.org/10.1145/3410530.3414386.

[91] J. Kim, B. McNally, L. Norooz, and A. Druin, “Internet search roles of adults in their homes,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’17, Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 4948–4959, ISBN: 9781450346559. DOI: 10.1145/3025453.3025572. [Online]. Available: https://doi.org/10.1145/3025453.3025572.

[92] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller, “Understanding in-video dropouts and interaction peaks inonline lecture videos,” in *Proceedings of the First ACM Conference on Learning @ Scale Conference*, ser. L@S ’14, Atlanta, Georgia, USA: Association for Computing Machinery, 2014, 31–40, ISBN: 9781450326698. DOI: 10.1145/2556325.2566237. [Online]. Available: https://doi.org/10.1145/2556325.2566237.

[93] K.-S. Kim and S.-C. J. Sin, “Use and evaluation of information from social media in the academic context: Analysis of gap between students and librarians,” *The Journal of Academic Librarianship*, vol. 42, no. 1, pp. 74–82, 2016, ISSN: 0099-1333. DOI: https : / / doi . org / 10 . 1016 / j . acalib . 2015 . 11 . 001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0099133315002438.

[94] S. Kita and S. Ide, “Nodding, aizuchi, and final particles in japanese conversation: How conversation reflects the ideology of communication and social relationships,” *Journal of Pragmatics*, vol. 39, no. 7, pp. 1242–1254, 2007.

[95] G. E. Knowlton and K. T. Larkin, "The influence of voice volume, pitch, and speech rate on progressive relaxation training: Application of methods from speech pathology and audiology," *Applied Psychophysiology and Biofeedback*, vol. 31, pp. 173–185, 2006. [Online]. Available: `https://api.semanticscholar.org/CorpusID:17505226`.

[96] A. M. Kring and A. H. Gordon, "Sex differences in emotion: Expression, experience, and physiology.," *Journal of personality and social psychology*, vol. 74, no. 3, p. 686, 1998.

[97] K. Kunze *et al.*, "The augmented narrative: Toward estimating reader engagement," in *Proceedings of the 6th Augmented Human International Conference*, ser. AH '15, Singapore, Singapore: Association for Computing Machinery, 2015, 163–164, ISBN: 9781450333498. DOI: `10.1145/2735711.2735814`. [Online]. Available: `https://doi.org/10.1145/2735711.2735814`.

[98] S. K. Kuttal, B. Ong, K. Kwasny, and P. Robe, "Trade-offs for substituting a human with an agent in a pair programming context: The good, the bad, and the ugly," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21, Yokohama, Japan: Association for Computing Machinery, 2021, ISBN: 9781450380966. DOI: `10.1145/3411764.3445659`. [Online]. Available: `https://doi.org/10.1145/3411764.3445659`.

[99] M. Kytö, I. Hirskyj-Douglas, and D. McGookin, "From strangers to friends: Augmenting face-to-face interactions with faceted digital self-presentations," in *Proceedings of the Augmented Humans International Conference 2021*, ser. AHs '21, Rovaniemi, Finland: Association for Computing Machinery, 2021, 192–203, ISBN: 9781450384285. DOI: `10.1145/3458709.3458954`. [Online]. Available: `https://doi.org/10.1145/3458709.3458954`.

[100] A. Lanitis, "Comparative evaluation of automatic age-progression methodologies," *EURASIP Journal on Applied Signal Processing*, 2008.

[101] S. Lübstorf and N. Lehmann-Willenbrock, "Are meetings really just another stressor? the relevance of team meetings for individual well-being," in Mar. 2020, pp. 47–69, ISBN: 978-1-83867-228-7. DOI: `10.1108/S1534-085620200000020003`.

[102] N. Lehmann-Willenbrock, J. A. Allen, and A. L. Meinecke, "Observing culture: Differences in u.s.-american and german team meeting behaviors," *Group Processes & Intergroup Relations*, vol. 17, no. 2, pp. 252–271, 2014. DOI: `10.1177/1368430213497066`. eprint: `https://doi.org/10.1177/1368430213497066`. [Online]. Available: `https://doi.org/10.1177/1368430213497066`.

[103] N. Lehmann-Willenbrock, J. A. Allen, and A. L. Meinecke, "Observing culture: Differences in u.s.-american and german team meeting behaviors," *Group Processes & Intergroup Relations*, vol. 17, no. 2, pp. 252–271, 2014, Visited on May 19, 2023. DOI: `10.1177/1368430213497066`. eprint: `https://doi.org/10.1177/1368430213497066`. [Online]. Available: `https://doi.org/10.1177/1368430213497066`.

[104] S. Lerner, “Projection boxes: On-the-fly reconfigurable visualization for live programming,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’20, Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–7, ISBN: 9781450367080. DOI: `10.1145/3313831.3376494`. [Online]. Available: `https://doi.org/10.1145/3313831.3376494`.

[105] H. Li, Z. Xing, X. Peng, and W. Zhao, “What help do developers seek, when and how?” In *2013 20th Working Conference on Reverse Engineering (WCRE)*, 2013, pp. 142–151. DOI: `10.1109/WCRE.2013.6671289`.

[106] C. Liu, J. Liu, and Y. Wei, “Scroll up or down? using wheel activity as an indicator of browsing strategy across different contextual factors,” in *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, ser. CHIIR ’17, Oslo, Norway: Association for Computing Machinery, 2017, pp. 333–336, ISBN: 9781450346771. DOI: `10.1145/3020165.3022146`. [Online]. Available: `https://doi.org/10.1145/3020165.3022146`.

[107] J. Makhlouf, Y. Arakawa, and K. Watanabe, “A privacy-aware browser extension to track user search behavior for programming course supplement,” in *Mobile and Ubiquitous Systems: Computing, Networking and Services*, T. Hara and H. Yamaguchi, Eds., Cham: Springer International Publishing, 2022, pp. 783–796, ISBN: 978-3-030-94822-1.

[108] M. Marron, “Log++ logging for a cloud-native world,” in *Proceedings of the 14th ACM SIGPLAN International Symposium on Dynamic Languages*, ser. DLS 2018, Boston, MA, USA: Association for Computing Machinery, 2018, 25–36, ISBN: 9781450360302. DOI: `10.1145/3276945.3276952`. [Online]. Available: `https://doi.org/10.1145/3276945.3276952`.

[109] D. Matsumoto, “Culture and nonverbal behavior,” *The SAGE handbook of nonverbal communication*, pp. 219–235, 2006.

[110] Y. Matsuyama, I. Akiba, S. Fujie, and T. Kobayashi, “Four-participant group conversation: A facilitation robot controlling engagement density as the fourth participant,” *Computer Speech & Language*, vol. 33, no. 1, pp. 1–24, 2015, Visited on May 19, 2023, ISSN: 0885-2308. DOI: `https://doi.org/10.1016/j.csl.2014.12.001`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0885230814001260`.

[111] L. M. Mauriello, S. S. Johnson, and J. M. Prochaska, “Meeting patients where they are at: Using a stage approach to facilitate engagement,” in *Practical Strategies and Tools to Promote Treatment Engagement*, W. O’Donohue, L. James, and C. Snipes, Eds. Cham: Springer International Publishing, 2017, pp. 25–44, Visited on May 19, 2023, ISBN: 978-3-319-49206-3. DOI: `10.1007/978-3-319-49206-3_3`. [Online]. Available: `https://doi.org/10.1007/978-3-319-49206-3_3`.

[112] K. McComas, C. Trumbo, and J. Besley, "Public meetings about suspected cancer clusters: The impact of voice, interactional justice, and risk perception on attendees' attitudes in six communities," *Journal of health communication*, vol. 12, pp. 527–49, Oct. 2007. DOI: 10.1080/10810730701508245.

[113] T. McDorman, "Implementing existing tools: Turning words into actions - decision-making processes of regional fisheries management organisations (rfmos)," *The International Journal of Marine and Coastal Law*, vol. 20, no. 3, pp. 423 –457, 2005, Visited on May 19, 2023. DOI: https://doi.org/10.1163/157180805775098595. [Online]. Available: https://brill.com/view/journals/estu/20/3/article-p423_4.xml.

[114] D. McDuff, R. Kaliouby, T. Senechal, M. Amr, J. Cohn, and R. Picard, "Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2013, pp. 881–888.

[115] A. Mehrabian, "Significance of posture and position in the communication of attitude and status relationships.," *Psychological bulletin*, vol. 71, no. 5, p. 359, 1969.

[116] D. Metaxa-Kakavouli, K. Wang, J. A. Landay, and J. Hancock, "Gender-inclusive design: Sense of belonging and bias in web interfaces," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, Montreal QC, Canada: Association for Computing Machinery, 2018, 1–6, ISBN: 9781450356206. DOI: 10.1145/3173574.3174188. [Online]. Available: https://doi.org/10.1145/3173574.3174188.

[117] O. Mohamad Nezami, M. Dras, L. Hamey, D. Richards, S. Wan, and C. Paris, "Automatic recognition of student engagement using deep learning and facial expression," in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds., Cham: Springer International Publishing, 2020, pp. 273–289, ISBN: 978-3-030-46133-1.

[118] H. Monkaresi, N. Bosch, R. A. Calvo, and S. K. D'Mello, "Automated detection of engagement using video-based estimation of facial expressions and heart rate," *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 15–28, 2017. DOI: 10.1109/TAFFC.2016.2515084.

[119] L.-P. Morency, C. Sidner, C. Lee, and T. Darrell, "Head gestures for perceptual interfaces: The role of context in improving recognition," *Artificial Intelligence*, vol. 171, no. 8-9, pp. 568–585, 2007.

[120] D. Morris, M. Ringel Morris, and G. Venolia, "Searchbar: A search-centric web history for task resumption and information re-finding," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08, Florence, Italy: Association for Computing Machinery, 2008, pp. 1207–1216, ISBN: 9781605580111. DOI: 10.1145/1357054.1357242. [Online]. Available: https://doi.org/10.1145/1357054.1357242.

[121] M. R. Morris, N. Moraveji, and D. Morris, “Supporting the social transfer of web search expertise,” in *CHI 2010 Workshop on the Next Generation of HCI and Education*, ACM, 2010. [Online]. Available: `https://www.microsoft.com/en-us/research/publication/supporting-social-transfer-web-search-expertise/`.

[122] J. E. Mroz, J. A. Allen, D. C. Verhoeven, and M. L. Shuffler, “Do we really need another meeting? the science of workplace meetings,” *Current Directions in Psychological Science*, vol. 27, pp. 484 –491, 2018. [Online]. Available: `https://api.semanticscholar.org/CorpusID:149475263`.

[123] K. Nagano, Y. Arakawa, and K. Yasumoto, “Trackthink: A tool for tracking a thought process on web search,” in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ser. UbiComp ’17, Maui, Hawaii: Association for Computing Machinery, 2017, pp. 681–687, ISBN: 9781450351904. DOI: `10.1145/3123024.3129267`. [Online]. Available: `https://doi.org/10.1145/3123024.3129267`.

[124] Y. Nakamura, Y. Matsuda, Y. Arakawa, and K. Yasumoto, “Waistonbelt x: A belt-type wearable device with sensing and intervention toward health behavior change,” *Sensors*, vol. 19, no. 20, 2019, Visited on May 19, 2023, ISSN: 1424-8220. DOI: `10.3390/s19204600`. [Online]. Available: `https://www.mdpi.com/1424-8220/19/20/4600`.

[125] K. Neureiter, M. Murer, V. Fuchsberger, and M. Tscheligi, “Hand and eyes: How eye contact is linked to gestures in video conferencing,” in *CHI ’13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’13, Paris, France: Association for Computing Machinery, 2013, pp. 127–132, ISBN: 9781450319522. DOI: `10.1145/2468356.2468380`. [Online]. Available: `https://doi.org/10.1145/2468356.2468380`.

[126] B. H. Nichols, S. Caplow, R. L. Franzen, L. R. McClain, L. Pennisi, and J. L. Tarlton, “Pandemic shift: Meeting the challenges of moving post-secondary environmental education online,” *Environmental Education Research*, vol. 28, no. 1, pp. 1–17, 2022, Visited on May 19, 2023. DOI: `10.1080/13504622.2021.2007220`. eprint: `https://doi.org/10.1080/13504622.2021.2007220`. [Online]. Available: `https://doi.org/10.1080/13504622.2021.2007220`.

[127] M. Niwa, K. Masai, S. Yoshida, and M. Sugimoto, “Investigating effects of facial self-similarity levels on the impression of virtual agents in serious/non-serious contexts,” in *Proceedings of the Augmented Humans International Conference 2023*, ser. AHs ’23, Glasgow, United Kingdom: Association for Computing Machinery, 2023, 221–230, ISBN: 9781450399845. DOI: `10.1145/3582700.3582721`. [Online]. Available: `https://doi.org/10.1145/3582700.3582721`.

[128] A. Ohnishi, K. Murao, T. Terada, and M. Tsukamoto, "A method for structuring meeting logs using wearable sensors," *Internet of Things*, vol. 5, pp. 140–152, 2019, ISSN: 2542-6605. DOI: `https://doi.org/10.1016/j.iot.2019.01.005`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2542660518301677`.

[129] OpenAI, *Introducing chatgpt*, 2022. [Online]. Available: `https://openai.com/blog/chatgpt`.

[130] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[131] H. H. Pham, H. Salmane, L. Khoudour, A. Crouzil, S. A. Velastin, and P. Zegers, "A unified deep framework for joint 3d pose estimation and action recognition from a single rgb camera," *Sensors*, vol. 20, no. 7, 2020, ISSN: 1424-8220. DOI: `10.3390/s20071825`. [Online]. Available: `https://www.mdpi.com/1424-8220/20/7/1825`.

[132] M. Poel, R. Poppe, and A. Nijholt, "Meeting behavior detection in smart environments: Nonverbal cues that help to obtain natural interaction," in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, IEEE, 2008, pp. 1–6. DOI: `10.1109/AFGR.2008.4813432`.

[133] M.-Z. Poh, D. J. McDuff, and R. W. Picard, "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation.," *Optics express*, vol. 18, no. 10, pp. 10 762–10 774, 2010.

[134] M. M. Rahman *et al.*, "Evaluating how developers use general-purpose web-search for code retrieval," in *Proceedings of the 15th International Conference on Mining Software Repositories*, ser. MSR '18, Gothenburg, Sweden: Association for Computing Machinery, 2018, 465–475, ISBN: 9781450357166. DOI: `10.1145/3196398.3196425`. [Online]. Available: `https://doi.org/10.1145/3196398.3196425`.

[135] A. Raiano, *Flowchart.js*, 2022. [Online]. Available: `http://adrai.github.io/flowchart.js`.

[136] K. Rayner, "The 35th sir frederick bartlett lecture: Eye movements and attention in reading, scene perception, and visual search," *Quarterly journal of experimental psychology*, vol. 62, no. 8, pp. 1457–1506, 2009.

[137] K. Ricanek and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 2006, pp. 341–345. DOI: `10.1109/FGR.2006.78`.

[138] K. Riedhammer, B. Favre, and D. Hakkani-Tür, "Long story short - global unsupervised models for keyphrase based meeting summarization," *Speech Commun.*, vol. 52, no. 10, 801–815, 2010, ISSN: 0167-6393. DOI: `10.1016/j.specom.2010.06.002`. [Online]. Available: `https://doi.org/10.1016/j.specom.2010.06.002`.

[139] P. Riehmann, M. Hanfler, and B. Froehlich, "Interactive sankey diagrams," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, IEEE, 2005, pp. 233–240.

[140] K. A. Rocca, "Student participation in the college classroom: An extended multidisciplinary literature review," *Communication education*, vol. 59, no. 2, pp. 185–213, 2010.

[141] S. Rogelberg, C. Scott, and J. E. Kello, "The science and fiction of meetings," *MIT Sloan Management Review*, vol. 48, pp. 18–21, 2007. [Online]. Available: `https://api.semanticscholar.org/CorpusID:51796277`.

[142] N. C. Romano and J. F. Nunamaker, "Meeting analysis: Findings from research and practice," in *Proceedings of the 34th annual Hawaii international conference on system sciences*, IEEE, 2001, 13–pp.

[143] N. C. Romano and J. F. Nunamaker, "Meeting analysis: Findings from research and practice," *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 13 pp.–, 2001. [Online]. Available: `https://api.semanticscholar.org/CorpusID:30777899`.

[144] C. Sadowski, K. T. Stolee, and S. Elbaum, "How developers search for code: A case study," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2015, Bergamo, Italy: Association for Computing Machinery, 2015, 191–201, ISBN: 9781450336758. DOI: `10.1145/2786805.2786855`. [Online]. Available: `https://doi.org/10.1145/2786805.2786855`.

[145] S. Samrose *et al.*, "Meetingcoach: An intelligent dashboard for supporting effective & inclusive meetings," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–13.

[146] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[147] S. Sarica and J. Luo, "Stopwords in technical language processing," *Plos one*, vol. 16, no. 8, e0254937, 2021.

[148] A. E. Scheflen, "The significance of posture in communication systems.," *Psychiatry*, vol. 27, pp. 316–31, 1964. [Online]. Available: `https://api.semanticscholar.org/CorpusID:31666916`.

[149] E. Schulte, N. Lehmann-Willenbrock, and S. Kauffeld, "Age, forgiveness, and meeting behavior: A multilevel study," *Journal of Managerial Psychology*, Jan. 2012. DOI: `10.1108/JMP-06-2013-0193`.

[150] M. Seligman and P Flourish, "A visionary new understanding of happiness and well-being," *M. Selligman, Flourish*, pp. 1–368, 2012.

[151] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[152] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[153] P. Sharma, S. Joshi, S. Gautam, V. Filipe, and M. C. Reis, “Student engagement detection using emotion analysis, eye tracking and head movement with machine learning,” *ArXiv*, vol. abs/1909.12913, 2019.

[154] S. Shin, J. Cho, and S.-W. Kim, “Jumple: Interactive contents for the virtual physical education classroom in the pandemic era,” in *Proceedings of the Augmented Humans International Conference 2021*, ser. AHs ’21, Rovaniemi, Finland: Association for Computing Machinery, 2021, 268–270, ISBN: 9781450384285. DOI: 10.1145/3458709.3458964. [Online]. Available: https://doi.org/10.1145/3458709.3458964.

[155] S. Shrivastava and V. Prasad, “Techniques to communicate in virtual meetings amidst the new normal ⋯ a consideration,” *Wutan Huatan Jisuan Jishu*, vol. 16, no. 6, pp. 73–92, 2020.

[156] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[157] H. Sloetjes and P. Wittenburg, “Annotation by category: ELAN and ISO DCR,” in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, N. Calzolari *et al.*, Eds., Marrakech, Morocco: European Language Resources Association (ELRA), 2008. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2008/pdf/208_paper.pdf.

[158] Y. Soneda, Y. Matsuda, Y. Arakawa, and K. Yasumoto, “M3b corpus: Multi-modal meeting behavior corpus for group meeting assessment,” in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC ’19 Adjunct, Visited on May 19, 2023, London, United Kingdom: Association for Computing Machinery, 2019, pp. 825–834, ISBN: 9781450368698. DOI: 10.1145/3341162.3345588. [Online]. Available: https://doi.org/10.1145/3341162.3345588.

[159] S. Song, J. Baba, J. Nakanishi, Y. Yoshikawa, and H. Ishiguro, “Mind the voice!: Effect of robot voice pitch, robot voice gender, and user gender on user perception of teleoperated robots,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA ’20, Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–8, ISBN: 9781450368193. DOI: 10.1145/3334480.3382988. [Online]. Available: https://doi.org/10.1145/3334480.3382988.

[160] L. Sprain and D. Boromisza-Habashi, “Meetings: A cultural perspective,” *Journal of Multicultural Discourses*, vol. 7, no. 2, pp. 179–189, 2012. DOI: 10.1080/17447143.2012.685743. eprint: https://doi.org/10.1080/17447143.2012.685743. [Online]. Available: https://doi.org/10.1080/17447143.2012.685743.

[161] N. Srivastava, “Using contactless sensors to estimate learning difficulty in digital learning environments,” in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC ’19 Adjunct, London, United Kingdom: Association for Computing Machinery, 2019, 399–403, ISBN: 9781450368698. DOI: 10.1145/3341162.3349312. [Online]. Available: https://doi.org/10.1145/3341162.3349312.

[162] N. Srivastava, E. Velloso, J. M. Lodge, S. Erfani, and J. Bailey, “Continuous evaluation of video lectures from real-time difficulty self-report,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’19, Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–12, ISBN: 9781450359702. DOI: 10.1145/3290605.3300816. [Online]. Available: https://doi.org/10.1145/3290605.3300816.

[163] N. Srivastava *et al.*, “Are you with me? measurement of learners’ video-watching attention with eye tracking,” in *LAK21: 11th International Learning Analytics and Knowledge Conference*, ser. LAK21, Irvine, CA, USA: Association for Computing Machinery, 2021, pp. 88–98, ISBN: 9781450389358. DOI: 10.1145/3448139.3448148. [Online]. Available: https://doi.org/10.1145/3448139.3448148.

[164] S. Y. Subramanya, K. Watanabe, A. Dengel, and S. Ishimaru, “Computer eyes need gender: Exploring the impact of gender as feature on facial expression emotion recognition for the inclusive world,” in *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems*, 2023 in review.

[165] D. Surani and H. Hamidah, “Students perceptions in online class learning during the covid-19 pandemic,” *International Journal on Advanced Science, Education, and Religion*, vol. 3, no. 3, pp. 83–95, 2020.

[166] H. Suzawa, K. Watanabe, M. Iwamura, K. Kise, A. Dengel, and S. Ishimaru, “Supporting smooth interruption in a video conference by dynamically changing background music depending on the amount of utterance,” in *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC ’22 Adjunct, Cambridge, United Kingdom: Association for Computing Machinery, 2023, 299–302, ISBN: 9781450394239. DOI: 10.1145/3544793.3560384. [Online]. Available: https://doi.org/10.1145/3544793.3560384.

[167] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[168] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, *Label studio: Data labeling software*, Open source software available from https://github.com/heartexlabs/label-studio, 2020-2022. [Online]. Available: `https://github.com/heartexlabs/label-studio`.

[169] A. Vargo *et al.*, "Learning cyclotron: An ecosystem of knowledge circulation," in *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC '22 Adjunct, Cambridge, United Kingdom: Association for Computing Machinery, 2023, 308–312, ISBN: 9781450394239. DOI: `10.1145/3544793.3560383`. [Online]. Available: `https://doi.org/10.1145/3544793.3560383`.

[170] R. Wampfler, S. Klingler, B. Solenthaler, V. R. Schinazi, M. Gross, and C. Holz, "Affective state prediction from smartphone touch and sensor data in the wild," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22, New Orleans, LA, USA: Association for Computing Machinery, 2022, ISBN: 9781450391573. DOI: `10.1145/3491102.3501835`. [Online]. Available: `https://doi.org/10.1145/3491102.3501835`.

[171] K. Watanabe, A. Dengel, and S. Ishimaru, "Metacognition-engauge: Real-time augmentation of self-and-group engagement levels understanding by gauge interface in online meetings," ser. AHs '24, Melbourne, Australia: Association for Computing Machinery, 2024 in review.

[172] K. Watanabe, H. K. Kucuk, S. Gonzales, A. Vargo, K. Kise, and S. Ishimaru, "Combining the knowledge of experienced programmers to extract useful web resources for solving programming tasks," in *Proceedings of the Asian HCI Symposium 2023*, ser. Asian CHI '23, New York, NY, USA: Association for Computing Machinery, 2023, pp. 22–27, ISBN: 9798400707612. DOI: `10.1145/3604571.3604575`. [Online]. Available: `https://doi.org/10.1145/3604571.3604575`.

[173] K. Watanabe, Y. Matsuda, Y. Nakamura, Y. Arakawa, A. Dengel, and S. Ishimaru, "Trackthink dashboard: Understanding student self-regulated learning in programming study," *Smart Learning Environments*, 2023 in review.

[174] K. Watanabe, Y. Matsuda, Y. Nakamura, Y. Arakawa, and S. Ishimaru, "How do programmers use the internet? discovering domain knowledge from browsing and coding behaviors," in *2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, IEEE, 2022, pp. 605–610.

[175] K. Watanabe, T. Sathyanarayana, A. Dengel, and S. Ishimaru, “Engauge: Engagement gauge of meeting participants estimated by facial expression and deep neural network,” *IEEE Access*, pp. 1–1, 2023. DOI: `10.1109/ACCESS.2023.3279428`.

[176] K. Watanabe, S. Tanaka, A. W. Vargo, K. Kise, and S. Ishimaru, “Trackthink camera: A tool for tracking facial and body information while web browsing,” in *AAGPW(at)AIED*, 2023. [Online]. Available: `https://api.semanticscholar.org/CorpusID:266598512`.

[177] K. Watanabe *et al.*, “Discaas: Micro behavior analysis on discussion by camera as a sensor,” *Sensors*, vol. 21, no. 17, 2021, Visited on May 19, 2023, ISSN: 1424-8220. DOI: `10.3390/s21175719`. [Online]. Available: `https://www.mdpi.com/1424-8220/21/17/5719`.

[178] R. W. White and D. Morris, “Investigating the querying and browsing behavior of advanced search engine users,” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’07, Amsterdam, The Netherlands: Association for Computing Machinery, 2007, pp. 255–262, ISBN: 9781595935977. DOI: `10.1145/1277741.1277787`. [Online]. Available: `https://doi.org/10.1145/1277741.1277787`.

[179] J. Williams, “Women at work,” in *Women vs Feminism*, Emerald Publishing Limited, 2017, pp. 31–47.

[180] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, “What do developers search for on the web?” *Empirical Software Engineering*, vol. 22, pp. 3149–3185, 2017.

[181] L. Xu, Z. T. Fernando, X. Zhou, and W. Nejdl, “Logcanvas: Visualizing search history using knowledge graphs,” in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR ’18, Ann Arbor, MI, USA: Association for Computing Machinery, 2018, 1289–1292, ISBN: 9781450356572. DOI: `10.1145/3209978.3210169`. [Online]. Available: `https://doi.org/10.1145/3209978.3210169`.

[182] Z. Ye *et al.*, “Detecting eye contact using wearable eye-tracking glasses,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, ser. UbiComp ’12, Pittsburgh, Pennsylvania: Association for Computing Machinery, 2012, 699–704, ISBN: 9781450312240. DOI: `10.1145/2370216.2370368`. [Online]. Available: `https://doi.org/10.1145/2370216.2370368`.

[183] M. Yoerger, J. Crowe, and J. A. Allen, “Participate or else!: The effect of participation in decision-making in meetings on employee engagement.,” *Consulting Psychology Journal: Practice and Research*, vol. 67, no. 1, p. 65, 2015.

[184] D. Yu and L. Deng, *Automatic speech recognition*. Springer, 2016, vol. 1.

[185] Z. Yu, Z. Yu, H. Aoyama, M. Ozeki, and Y. Nakamura, “Capture, recognition, and visualization of human semantic interactions in meetings,” Mar. 2010, pp. 107–115. DOI: `10.1109/PERCOM.2010.5466987`.

[186] F. Zermiani, A. Bulling, and M. Wirzberger, “Mind wandering trait-level tendencies during lecture viewing: A pilot study,” in *2022 Symposium on Eye Tracking Research and Applications*, ser. ETRA ’22, Seattle, WA, USA: Association for Computing Machinery, 2022, ISBN: 9781450392525. DOI: 10.1145/3517031.3529241. [Online]. Available: https://doi.org/10.1145/3517031.3529241.

[187] Z. Zhai, X. Chen, Y. Zhao, L. Zhao, J. Qian, and J. Wu, “Smartcamera: Realtime video stream-oriented action recognition platform in edge environment,” in *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, ser. UbiComp ’21, Virtual, USA: Association for Computing Machinery, 2021, 88–89, ISBN: 9781450384612. DOI: 10.1145/3460418.3479303. [Online]. Available: https://doi.org/10.1145/3460418.3479303.

[188] L. Zhang *et al.*, “End-to-end automatic pronunciation error detection based on improved hybrid ctc/attention architecture,” *Sensors*, vol. 20, no. 7, 2020, ISSN: 1424-8220. DOI: 10.3390/s20071809. [Online]. Available: https://www.mdpi.com/1424-8220/20/7/1809.

[189] X. Zhang, Y. Sugano, and A. Bulling, “Everyday eye contact detection using unsupervised gaze target discovery,” in *Proceedings of the 30th annual ACM symposium on user interface software and technology*, 2017, pp. 193–203.

[190] R. Zhao, V. Li, H. Barbosa, G. Ghoshal, and M. E. Hoque, “Semi-automated 8 collaborative online training module for improving communication skills,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, 2017, Visited on May 19, 2023. DOI: 10.1145/3090097. [Online]. Available: https://doi.org/10.1145/3090097.

[191] B. J. Zimmerman and D. H. Schunk, *Self-regulated learning and academic achievement: Theoretical perspectives*. Routledge, 2001.

Own Publications

Journal Paper

1. Ko Watanabe, Yuki Matsuda, Yugo Nakamura, Yutaka Arakawa, Andreas Dengel, Shoya Ishimaru. TrackThink Dashboard: Understanding student self-regulated learning in programming study. Smart Learning Environment 10, 42 (2023), in review.

2. Ankur Bhatt, Ko Watanabe, Andreas Dengel, Shoya Ishimaru. Appearance-Based Gaze Estimation with Deep Neural Networks: From Data Collection to Evaluation. International journal on Activity and behavioral Computing, 2023, in press.

3. Ko Watanabe, Tanuja Sathyanarayana, Andreas Dengel and Shoya Ishimaru. EnGauge: Engagement Gauge of Meeting Participants Estimated by Facial Expression and Deep Neural Network. IEEE Access, pp. 52886-52898, 2023.

4. Chenhao Chen, Yutaka Arakawa, Ko Watanabe and Shoya Ishimaru. Quantitative Evaluation System for Online Meetings Based on Multimodal Microbehavior Analysis. Sensors and Materials 34 (8), pp. 3017–3027, 2022.

5. Ko Watanabe, Yusuke Soneda, Yuki Matsuda, Yugo Nakamura, Yutaka Arakawa, Andreas Dengel and Shoya Ishimaru. DisCaaS: Micro Behavior Analysis on Discussion by Camera as a Sensor. Sensors 21 (17), p. 5719, 2021.

International Conference Papers

1. Sahana Yadnakudige Subramanya, Ko Watanabe, Andreas Dengel, Shoya Ishimaru. Exploring the Impact of Gender as a Feature on Facial Expression Emotion Recognition. In 2024 12th International Conference on Affective Computing and Intelligent Interaction (ACII), 2024, in review.

2. David Dembinsky, Ko Watanabe, Andreas Dengel, Shoya Ishimaru. Eye Movement in a Controlled Dialogue Setting. In 2024 Symposium on Eye Tracking Research and Applications (ETRA ’24), 2024, in press.

3. Ko Watanabe, Andreas Dengel, Shoya Ishimaru. Metacognition-EnGauge: Real-time Augmentation of Self-and-Group Engagement Levels Understanding by Gauge Interface in Online Meetings. In the Augmented Humans International Conference (AHs ’24), 2024, in press.

4. Ko Watanabe, Andreas Dengel and Shoya Ishimaru. Accelerating Knowledge Transfer by Sensing and Actuating Social-Cognitive States. In Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing

and Proceedings of the 2023 ACM International Symposium on Wearable Computers (UbiComp/ISWC '23 Adjunct), 2023.

5. Ko Watanabe, Seiya Tanaka, Koichi Andrew amd Kise Vargo and Shoya Ishimaru. TrackThink Camera: A Tool for Tracking Facial and Body Information while Web Browsing. In AIED' 23 Workshop: Automated Assessment and Guidance of Project Work, 2023.

6. Ko Watanabe, Hacer Kübra Kücük, Sarah Gonzales, Andrew Vargo, Koichi Kise and Shoya Ishimaru. Combining the Knowledge of Experienced Programmers to Extract Useful Web Resources for Solving Programming Tasks. In Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23 Asian CHI Symposium), 2023.

7. Haruki Suzawa, Ko Watanabe, Masakazu Iwamura, Koichi Kise, Andreas Dengel and Shoya Ishimaru. "Supporting Smooth Interruption in a Video Conference by Dynamically Changing Background Music Depending on the Amount of Utterance". In Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '22 Adjunct), pp. 299–302, 2022.

8. Ko Watanabe, Yuki Matsuda, Yugo Nakamura, Yutaka Arakawa and Shoya Ishimaru. How do Programmers Use the Internet? Discovering Domain Knowledge from Browsing and Coding Behaviors. In 2022 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), pp. 605-610, 2022.

9. Jihed Makhlouf, Yutaka Arakawa and Ko Watanabe. A Privacy-aware Browser Extension to Track User Search Behavior for Programming Course Supplement. In The Fourth International Workshop on Mobile Ubiquitous Systems, Infrastructures, Communications and AppLications (MUSICAL) In conjunction with Mobiquitous 2021, 2021.

Papers Not Related to the Thesis or with My Small Contribution

1. Riku Higashimura, Andrew Vargo, Ko Watanabe, Motoi Iwata, Shoya Ishimaru, Andreas Dengel, Koichi Kise. Estimating Unknown Words on Smartphones Via User Behavior. In Proceedings of the 26th International Conference on Mobile Human-Computer Interaction (Mobile-HCI '24), 2024, in review.

2. Ryugo Morita, Hitoshi Nishimura, Ko Watanabe, Andreas Dengel and Jinjia Zhou. Edge-based Denoising Image Compression. In Proceedings of the 2024 32nd European Signal Processing Conference (EUSIPCO), 2024, in review.

3. Kanta Yamaoka, Ko Watanabe, Koichi Kise, Andreas Dengel and Shoya Ishimaru. Experience is the Best Teacher: Personalized Vocabulary Building Within the Context of

Instagram Posts and Sentences from GPT-3. In Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '22 Adjunct), pp. 313–316, 2022.

4. Andrew Vargo, Motoi Iwata, Mathilde Hutin, Sofiya Kobylyanskaya, Ioana Vasilescu, Olivier Augereau, Ko Watanabe, Shoya Ishimaru, Benjamin Tag, Tilman Dingler, Koichi Kise, Laurence Devillers and Andreas Dengel. Learning Cyclotron: An Ecosystem of Knowledge Circulation. In Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '22 Adjunct), pp. 308-312, 2022.

5. Shoya Ishimaru, Ko Watanabe, Nicolas Großmann, Carina Heisel, Pascal Klein, Yutaka Arakawa, Jochen Kuhn and Andreas Dengel. HyperMind Builder - Pervasive User Interface to Create Intelligent Interactive Documents. Proc. UbiComp 2018 Adjunct, pp. 357-360, October 2018.

Curriculum Vitae

Summary

Ko Watanabe is a researcher at the German Research Center for Artificial Intelligence (DFKI). His research interest is to invent new technologies to enhance knowledge transfer in human-computer interaction. He received B.E. degree in Engineering from Tokyo University of Agriculture and Technology and M.E. in Information Science from Nara Institute of Science and Technology in 2017 and 2019. He then worked at the company in Japan for two years and started his Ph.D. in Engineering at the RPTU Kaiserslautern-Landau in 2021.
Ko is also a software developer. He likes programming and applying systems in real-life scenes or actual products. As an internship/part-time engineer, he worked for many IT companies, including Allesgood, Rakuten, and DeNA. In 2018, he received a scholarship from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) and Japan Science and Technology (JST). He gave a number of invited talks at universities.

Background

Human-Computer Interaction, Learning Analytics, Affective Computing, and Pattern Recognition

Work Experience

Full-time PhD Researcher, RPTU Kaiserslautern-Landau	*March 1st 2021 – June 30th 2024*
Full-time Software Engineer, Pococha, DeNA Co., Ltd	*April 2019 – Feburuary 2021*

Education

Doctor of Engineering, RPTU Kaiserslautern-Landau	*March 1st 2021 – November 2024*
Master of Engineering, Nara Institute of Science and Technology	*April 2017 – March 2019*
Bachelor of Engineering, Tokyo University of Agriculture and Technology	*April 2013 – March 2017*

Funds

Representative, JST CREST AIP Challange, 1.00 M JPY	*2021-2022*
Representative, MEXT Tobitate Scholarship, 1.16 M JPY	*2017-2018*

Selected Part-Time Job and Service

Visiting Researcher, Osaka Metropolitan University	*2021 – current*

Selected Publications

Ko Watanabe, Shoya Ishimaru and Andreas Dengel. "EnGauge: Engagement Gauge of Meeting Participants Estimated by Facial Expression and Deep Neural Network". In IEEE Access, vol. 11, pp. 52886-52898, 2023. *Journal Paper*

Ko Watanabe, Yusuke Soneda, Yuki Matsuda, Yugo Nakamura, Yutaka Arakawa, Andreas Dengel, and Shoya Ishimaru. "DisCaaS: Micro Behavior Analysis on Discussion by Camera as a Sensor". In Sensors 21, no. 17: 5719, 2021. *Journal Paper*

Ko Watanabe, Andreas Dengel, Shoya Ishimaru. "Metacognition-EnGauge: Real-time Augmentation of Self-and-Group Engagement Levels Understanding by Gauge Interface in Online Meetings". In the Augmented Humans International Conference (AHs '24), 2024. *Conference Paper*