Facial Expression Recognition in Improving User Experience

Master's Thesis in Technology
University of Turku
Department of Computing
Software Engineering
2026
Sara Keskilohko

Acknowledgements

Finishing this thesis wouldn't have been possible without the people who stood by me through my years at the University of Turku. I want to thank my supervisor and friend Erno Lokkila for supporting me in my work through my years of studying. Their guidance with my thesis has been priceless. I am grateful to all who took part in the study and helped me reach this goal. I am also deeply thankful for the people closest to me, friends and family, for their continuous support. Sincere thanks are given for the guild of computer engineering Digit ry for the best years of my life. This thesis is dedicated to all of you, you know who you are.

Turku, April 2026

Sara Keskilohko

UNIVERSITY OF TURKU
Department of Computing

SARA KESKILOHKO: Facial Expression Recognition in Improving User Experience

Master's Thesis in Technology, 98 p., 13 app. p.
Software Engineering
April 2026

The utilization of AI in emotion-aware systems has been a rising trend in the latest years. With emotion-aware systems and interfaces user experience can be improved, and different modalities can help in achieving this improvement. Facial expression recognition is an AI based technology, and its output is a type of signal that can be used in user interface adaptation.

The research methods for this thesis were a literature review and an empirical study utilizing an adaptive web application with two integrated facial expression recognition models. The study collected data on detected facial expressions, accuracy in performing tasks in the application, and on how user interface adaptation affected the performance accuracy of the participants and the facial expression recognition models. 25 participants were included in the study, and after completing the tasks in the application the participants filled a survey reflecting on their emotions throughout the experiment.

The literature review and conducted study show that emotion-aware adaptive applications can be designed successfully with cloud-based APIs, but there is still a need for more research in the field. The performance of the two facial expression recognition models varied greatly, and improvements in the abilities of detecting faces and recognizing more emotions is something to research more in the future. An adaptive user interface improved the user experience of around 20% of the participants, and 10% stated that the user interface was the main factor for positive emotions felt through the experiment.

Keywords: facial expression recognition, user experience, adaptive user interface, real-time emotion detection, AWS Rekognition, Google Vision

UNIVERSITY OF TURKU
Department of Computing

Sara Keskilohko: Facial Expression Recognition in Improving User Experience

Master's Thesis in Technology, 98 p., 13 app. p.
Software Engineering
April 2026

Tekoälyn hyödyntäminen tunteita tunnistavissa järjestelmissä on ollut nousussa viime vuosina. Tunteita tunnistavat järjestelmät ja tunteiden tunnistamista hyödyntävät käyttöliittymät voivat luoda paremman käyttökokemuksen, ja tämä voidaan saavuttaa hyödyntämällä erilaisia antureita ja kanavia aistikokemusten välittämiseen. Ilmeiden tunnistus on teknologia, joka hyödyntää erityisesti tekoälyä, näihin erikoistuneiden mallien antamia tuloksia voidaan hyödyntää käyttöliittymien mukauttamiseen.

Tämän opinnäytetyön tiedonhakumenetelmät olivat kirjallisuuskatsaus sekä empiirinen tutkimus, jossa käytettiin itseohjelmoitua web-sovellusta. Web-sovelluksessa hyödynnettiin kahta eri ilmeiden tunnistukseen kykenevää tekoälymallia. Työn tutkimuksessa kerättiin dataa mallien tunnistamista ilmeistä ja niiden oikeellisuudesta, sovelluksesta löytyvien tehtävien suorittamisen tarkkuudesta, sekä mukautuvan käyttöliittymän vaikutuksista tehtävien suorittamiseen sekä ilmeidentunnistusmallien toimivuuteen. Tutkimukseen osallistui 25 henkeä, ja web-sovelluksen tehtävien suorittamisen jälkeen osallistujat täyttivät kyselyn, jossa he kertoivat tunteista ja tuntemuksista, joita he kokivat tutkimuksen tehtävien suorittamisen aikana.

Kirjallisuuskatsaus sekä tutkimus osoittavat, että sovelluksia, jotka pystyvät tunnistamaan tunteita, pystytään suunnittelemaan ja ohjelmoimaan pilvipohjaisten API:en avulla, mutta tällä alalla tehdyt tutkimukset ovat vähäisiä, minkä vuoksi näyttöä hyödyllisyydestä ja parhaista tavoista hyödyntää teknologiaa tarvitaan lisää. Tutkimuksessa käytettyjen mallien tulokset ja suorituskyky vaihtelivat keskenään paljon, erityisesti oikeiden tunteiden tunnistamisessa kumpikaan malli ei suoriutunut täydellisesti. Lisää tutkimusta tarvitaan siitä, miten mallit saadaan tunnistamaan kasvot ja tunteet oikein, sekä siitä, miten ne pystyvät jatkossa tunnistamaan useampia tunteita. Mukautuva käyttöliittymä paransi käyttökokemusta noin 20%:lla osallistujia, ja 10% osallistujista sanoi käyttöliittymän vaikuttaneen heihin eniten positiivisesti tutkimuksen aikana.

Keywords: ilmeiden tunnistus, käyttökokemus, mukautuva käyttöliittymä, tunteiden tunnistaminen reaaliajassa, AWS Rekognition, Google Vision

Contents

List of Figures

List of Tables

List of acronyms

AI Artificial Intelligence

API Application Programming Interface

CNN Convolutional Neural Networks

DFER Dynamic Facial Expression Recognition

FER Facial Expression Recognition

HCI Human Computer Interaction

ML Machine Learning

NN Neural Networks

UI User Interface

UX User Experience

1 Introduction

The use of artificial intelligence (AI) has been on the rise for several years, but during the most recent years the technology has taken huge steps forward. This has made artificial intelligence more accessible for everyone. The average person can see and use something made or edited by artificial intelligence multiple times a day, sometimes without even noticing it. With this in mind it's beneficial to turn the gaze towards utilizing the technology in an efficient way that is still safe and ethical. The field of information technology has used AI for different tasks for many years, but there's still much to learn on how to utilize it in the best way for a certain purpose.

AI is used in current interfaces and software design more often. Most cellphones come with a built in AI assistant, websites offer AI summaries and AI chat bots can be used anywhere. Besides AI recommendation systems have been used in technology and applications for multiple years. They have been introduced as early as the 1990's, and their aim is to offer personalized recommendations to the users [1]. The need for filtering options and content has been a part of the industry for two decades. Recommendation systems, or sometimes called recommender systems, are tools for filtering content and providing the user with suggestions that might please them [2]. For example Netflix still uses a recommendation system for its content, although it no longer is just an algorithmic recommendation system, but

rather developed with the use of deep learning [3].

Usability and user experience (UX) has been a part of system design for the last decades, and in the most recent years it has become a marketing chip and a priority for IT products of all kind [4]. Products are designed to be easy to use and intuitive. Since this has become a rising trend it would be beneficial to investigate how UX can be improved with automatization. AI is often utilized for automating tasks in all fields, so it would be natural to utilize it to also automatize adaptation in interfaces.

The ways that AI could be used for improving UX have already been discussed for some years. It is currently used in for example designing an interaction system with inherently good UX [5]. Samples of user data and graphical user interface (UI) elements can be used as the training set for the AI used for designing tasks, and it can therefore enhance the design progress [5]. Still good user experience doesn't stem from just inputting all heuristics for UX into a language model and expecting it to succeed.

Real-time human computer interaction (HCI) is a topic that is often mentioned when discussing adaptive interfaces and adaptation of systems. Human-computer interaction is fundamentally people and machines interacting with each other, and it's been a topic for longer than artificial intelligence has [6]. These systems can be as simple as a computer, a keyboard and a mouse, so artificial intelligence is not needed for basic HCI. Deep learning has changed the focus toward HCI being a real-time adaptive system [6]. Therefore also emotion based interaction has been researched more. HCI often aims to use multi-modal input, meaning that different ways of inputting information is used with the system [6]. This means that the system can receive input with for example a camera and by voice recognition. Real-time interaction happens when the system can in real-time process the data that it receives and can respond to that. Even real-time systems have some latency, but a

system can be called real-time if its latency is small enough.

Personalization is a part of adaptivity, where the user interface or the experience is personalized for a specific individual [7]. It uses data about the individual to adapt to their needs. Adaptivity and personalization has already been utilized in health care and in education [8]. Personalization still hasn't reached the every-day user interface design, and personalization is more often linked to content filtering than it is to UI adaptation. Especially contextual UI adaptation could have many applications once it's been researched more [8].

The need for personalization and adaptation as a whole comes from different needs for the same application or interface. UX has best practices and several heuristic models for a successful use experience, such as the famous Nielsen's ten heuristics for software design, upon which more heuristic evaluation systems have been based [9]. Still a list set in stone doesn't guarantee a positive user experience for every user. This is where personalization and adaptive user interfaces can improve the experience. Different users of different applications need different levels of support, and even though it is said in Nielsen's heuristics, that the system has to be the appropriate level for the user, the users themselves might differ in the support level that is needed. Since artificial intelligence has been proved as a useful tool for optimization it could also be used to optimize the UX for the specific user.

Facial expression recognition (FER) systems have gained momentum, and they are being used in different applications and devices [10]. Facial expression recognition systems are HCI systems that use a camera as the input method. Facial recognition has utilized for emotion detection through algorithms analyzing photographs, but the technology for real-time recognition especially from video footage is a newer trend [11]. The more traditional FER systems have remained solely dependent on feature recognition, and while current FER systems also utilize this technology, it

faces issues with for example label noise [10]. For example medical professionals are utilizing facial expression recognition already in understanding and monitoring well-being [12]. Emotion recognition systems are often multi-modal, but for facial expression recognition a camera is the general choice.

Current challenges with facial expression recognition are about utilizing it in an ethical way, and on getting accurate readings. Throughout the recent years it's been discussed that FER models are often trained with data that could be biased [13]. In addition to biased datasets FER is a new technology, and privacy issues have been discussed in detail, especially since a picture or video feed is always needed for facial expression recognition [13].

Artificial intelligence can be used as an assistant working to improve the experience. When combined with FER it's possible to create systems that respond to the detected facial expressions in a way that is beneficial for the user of the interface. Here the emotional state could be declared as a factor in for example CSS design.

Real-time adaptive user interfaces could be utilized in creating a more positive user experience, as well as personalize the user interface to respond to what the user needs from it. This can make the interface easier to use for those who would otherwise need help navigating it. With real-time adaptation frustration could even be decreased and productivity increased.

1.1 Research Objectives

The thesis uses a systematic literature review and a case study as its main research methods. The case study aimed to build a real-time adaptive web application that utilizes facial expression recognition systems as its input. Through this study the goal was to evaluate how accurately the current systems work and whether the

adaptation of the user interface affects the user experience. The facial expression recognition systems used for this study are cloud based artificial intelligence driven application programming interfaces (APIs) made by Amazon Web Services and Google.

The built system utilized Amazon Web Services' Rekognition and Google's Cloud Vision API for facial expression recognition. The system was a web based application built with React, Node.js and Express. Artificial intelligence Chat GPT 5.3 was utilized as help when programming the application. The application used a web camera and took frames from the feed that were sent to both of the FER APIs. Based on the readings from the FER systems a stress level was determined every 6 seconds. The changes in the stress level prompted the UI to change, and the aim was to make the UI more pleasing the more stressed the participant looked. The UI included 12 different levels of support, and the support level could only increase during the experiment run. After completing the tasks on the application the participants were asked to answer a survey on how they felt and what caused the emotions. These methods were used to find does a comforting UI reduce stress felt by the participants, while the tasks in the application got more difficult. The raw numerical data on the emotions detected by the FER systems was also saved, so that the accuracy of the FER readings could be studied by matching them with the participants' survey answers.

Artificial intelligence was used as assistance in programming the experiment application, in writing the data analysis scripts, and in formatting tables and appendices in LaTeX for the thesis. AI-assisted outputs were reviewed, edited, and verified by the author. The specific language models used were Chat GPT 5.3 by OpenAI and Claude Sonnet 4.6 by Anthropic.

The research questions for this thesis are the following.

RQ1: How can facial expression recognition technology and AI be used to create a positive user experience?

RQ2: How accurately do the existing technologies distinguish emotions?

RQ3: What constraints does the existing technology pose in recognizing facial expressions?

RQ4: How does an adaptive interface affect the user experience?

1.2 Scope and Limitations

The literature used for this thesis has been carefully selected. All references are five years old at the most, since the field of AI is growing rapidly, therefore requiring current and accurate literature.

This thesis doesn't evaluate a self made FER system, but compares two well known systems and their abilities in a real life setting. Attributes like gender, etc, are not tracked with the FER systems even though it would be possible. The evaluation of the two FER providers gives insight into what technologies and models are currently more accurate and realistic for real-time facial expression recognition.

The case study in this thesis only contained the data from 20 participants. Therefore not all results might not be generalizable to larger focus groups. The test setting was controlled and partially time limited, and the experiment procedure was the same for all participants, no matter the background of the participants. The participants weren't asked to describe their emotional state before starting the experiment, and it was possible that emotional state of the participant throughout the experiment was affected by the emotions felt previous to the experiment. Therefore some emotions felt during the experiment might've not been completely related to the exercises

themselves.

In the experiment the only modality that was used was a camera. The only indications about stress levels were therefore received through pictures. The stress was calculated purely from facial expressions. For example heart rate monitors could have been an addition to monitoring how stressed the participants were feeling. Some people don't show their stress on their face, and some might avoid being expressive in an experiment setting like this.

The experiment was conducted with only two providers, and even though the providers are notable companies worldwide there are many other successful companies providing facial expression recognition services. For this thesis there was no need to evaluate more than two systems by different providers. For more accurate view on the entirety of facial expression recognition systems' performance more studies should be conducted. No self-trained datasets or language models were evaluated in the thesis.

1.3 Structure

Chapter 2 is a literary review on the topics behind facial expression recognition models and adaptive user interfaces. It also discusses the basics of artificial intelligence that is needed for working FER models. Research question 1 is answered in this chapter.

Chapter 3 introduces the application that was built for the thesis. It explains the application's design and architecture, and how the system works. The two facial expression recognition APIs used for the application are also introduced here more deeply. The APIs are Amazon Web Services Rekognition and Google Cloud Vision API. This is the system that was built to investigate research questions 2, 3 and 4.

The experiment that the application was built for is introduced in Chapter 4. The experiment setting as well as the data analysis based on the data from the experiment is reviewed in this chapter as well. The experiment setting is needed for research questions 2, 3 and 4.

The fifth chapter introduces and evaluates the findings from the experiment. This entails evaluation on the performance on the two FER systems and correlations between emotions and test performances. The research questions 2 and 3 are answered in this chapter.

The final chapter before conclusion discusses the findings and their rationality. It also evaluates whether or not the findings correspond with what was found by the literature review. This chapter answers research question 4. It also suggests how this study could be taken forward.

2 Using Artificial Intelligence in User Interfaces

Artificial intelligence (AI) is the product of many technologies, such as machine learning, neural networks, human-computer interaction and decision making [14][15]. Different kinds of algorithms have been used for a long time to emulate human behavior but with current technologies artificial intelligence has been able to gain huge traction. It is being used in all branches of science, medicine, humanistic studies and education. Even though its history can be traced back to the 1950's, the field of AI has taken huge leaps during the last few years [15]. Now, in the year 2025, AI applications are everywhere, and many people utilize generative AI on the daily, and it's gained popularity among personal use as well as use in business environments. The reason for using AI can range anywhere from reducing personnel costs and following the trends to using it to make completing tasks more efficient and reduce human mistakes [15].

Before artificial intelligence as we now know it, expert systems were used. Expert systems are by definition experts regarding specific information, and AI itself can be considered as an expert system since it contains vast amounts of expert knowledge on many topics.

Artificial intelligence itself has many subsets, one of which is machine learning (ML) [16]. Artificial intelligence itself can be be purely rule-based mathematical models, although nowadays when discussing about AI people often mean machine learning algorithms or large language models, such as ChatGPT. The term "AI" is just easily linked to these actual terms in peoples minds [17]. These rule-based AI applications work based on a set of rules, for example "if - then" statements, and it doesn't adapt over time [17].

What makes machine learning different is its ability to learn and improve itself [15]. It aims to match human behavior and it learns from its mistakes. Essential for machine learning algorithms are neural networks. These artificial neural networks are build to mimic the way neural networks work in the human brain, therefore aiming to match the way a human thinks and makes choices [16]. Some of the most common techniques in machine learning are decision trees and support vector machines [17]. Machine learning algorithms are trained with large data sets that are labeled. With the correct labels or annotations on the data the ML algorithms can acquire a nearly perfect ability of pattern recognition [18]. ML algorithms are initially pattern recognition machines [16]. Once they have been trained sufficiently they learn the ability to predict the outcome based on the input they receive [16].

The labeling for a dataset can be done with different types of learning methods, such as supervised learning, unsupervised learning or transfer learning. For example in facial expression recognition the learning is often supervised or transfer learning. Supervised learning is done with a human that labels and annotates the data that is fed to the ML algorithm [19]. The data has been carefully labeled, and when the algorithm has been taught sufficiently, it should be able to make the correct predictions on similar data in the future. The training in supervised learning consists of training datasets that are annotated and testing datasets that are not [19]. The

test sets are used to see can the algorithm already correctly make predictions. One way to label is to make class labels, such as "sad" or "angry" that can be used in classification in facial expression recognition as well.

Transfer learning utilizes already trained models to complete a different task, that what they initially were trained for [20]. In this case the model is already taught on certain topics, and it can for example do some classifications or pattern recognition. With transfer learning it is fed different kind of data and knowledge on topics, and this way it is tuned to the new task [20]. Transfer learning is useful if there is not a lot of actual data to specifically train it with [20].

Neural networks (NN) that are used in machine learning consist of layers, which consist of artificial neurons. These artificial neurons are made to mimic the way neurons function in the human brain [21]. The different amount of layers and how the neurons are located on the layers compose different neural network models. The layers that all neural networks have are the input layer and the output layer. Between them there are a selected number of hidden layers [21]. The accuracy and performance of the model can be improved with additional hidden layers to the model. The layers consisting of nodes of neurons are connected in a web through which the information is processed [21]. When there are several hidden layers in the model it is called a deep neural network [21]. Deep neural networks are used in deep learning, which is a subset of machine learning [16]. A simplified neural network model is demonstrated in Figure 2.1.

The different connections between the nodes of neurons have assigned weights, that affect the output. These weights are summed and the sum is passed onto the activation function, which is responsible for producing the output [21]. This is how ML algorithms are trained with datasets. This is demonstrated in Figure 2.2.

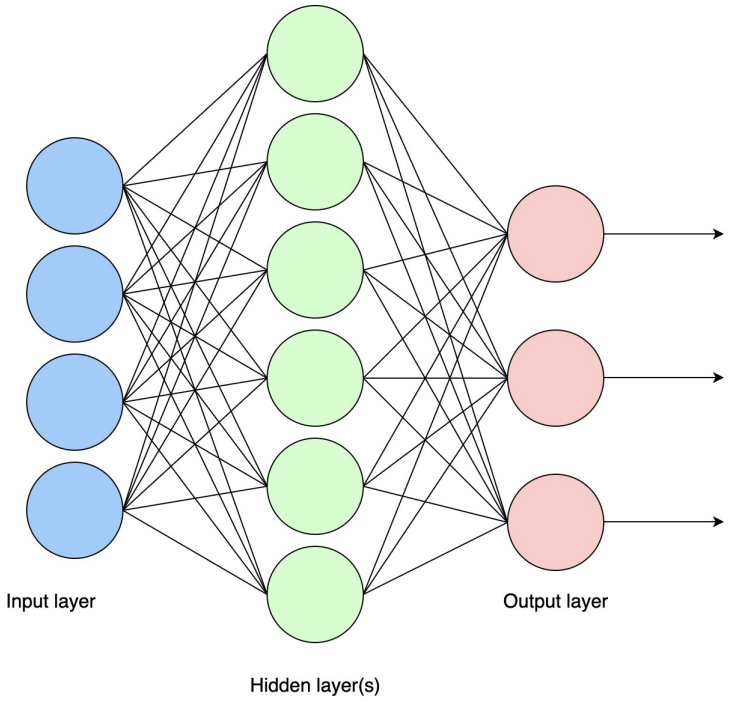


Figure 2.1: A simplified model of neural network layers

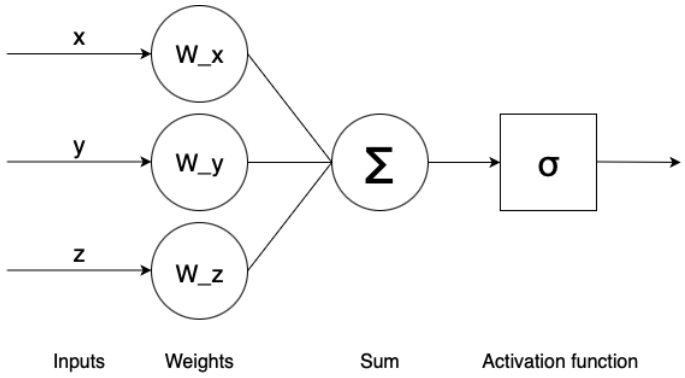


Figure 2.2: Weighted sum in producing output

There are several neural network models that have gained popularity. Some of the most commonly used are the following:

- Multi-Layer Perceptron (MLP)
- Self-Organizing Map (SOM)
- Convolutional Neural Network (CNN)

The multi-layer perceptron is the most widely used model, but choosing the correct model depends on the tasks the AI should accomplish [22]. For example for facial expression recognition the most used models are convolutional neural networks and MLP [23][24]. The constructs under which facial expression recognition falls is visually demonstrated in Figure 2.3.

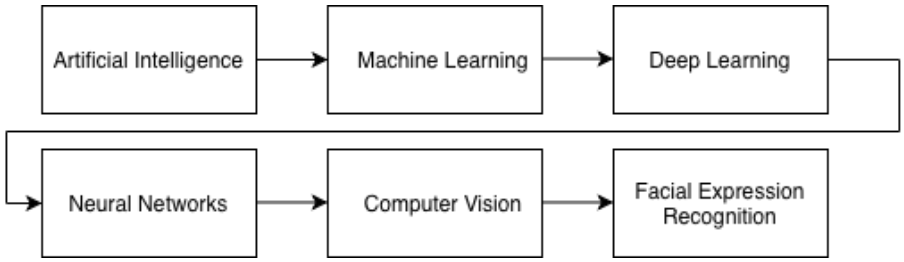


Figure 2.3: Hierarchy for facial expression recognition

Artificial intelligence can be utilized in many different parts of the process of interface designing. It can be used to automate tasks and scripts, modify and generate content, interact with the user and analyze the user's behavior [18]. AI in user interfaces can be made visible to the user for example through AI chat bots and text-to-speech expansions using AI, but the uses for it go much further than just what the user can see. The creator of the user interface can use AI in the design process, it can be made available for the user on the user interface, it can be utilized to study user amounts and user behavior and even to design new versions of the user interface based on what was found through the users' behavior.

2.1 Usability

Usability is core principle of smooth and functional user interfaces [25]. It affects the user experience and can help the user get more out of the interface and therefore the whole program or system. Although the importance of usability has been debated over the years the consensus nowadays seems to be that usability should always

be treated as an important factor in user experience [25]. One way to impact UIs and software development as a whole is automatization. Using automatization for improving the user experience, however, hasn't been a trend until quite recently [26]. AI is especially beneficial when talking about automating tasks [27]. Automatization can be a big asset to more efficient ways of improving user experience, and it can also be applied to analyzing data on user performance, which often can go hand in hand with good usability and user experience. Often user experience can be split into four different properties, which are the aforementioned usability in addition to accessibility, credibility and findability [26]. Out of these four usability describes how easy the user interface or system is to navigate.

Usability can be defined with the ISO 9241 standards set by the International Organization for Standardization. The standardization ISO 9241-11 from 2018 states that usability consists of of three main factors, that are effectiveness, efficiency and satisfaction [28]. ISO/IEC 25010 from 2008 lists factors for usability to be accessibility, operability, learnability, user error protection, UI aesthetics and appropriateness recognizability [28]. An older standardization from 2006, ISO 9241-110, states that the key interaction principles are suitability to tasks, controllability, customization, learnability, conformity with expectations, self-descriptiveness, user error awareness and user engagement [29]. It can be seen that the standards are updated to better respond to improving technologies, but the core values stay the same. These standards shape how user interfaces are designed and realized to create a positive user experience.

Usability itself can be measured in multiple different ways, and the metrics are chosen based on what the application that is inspected contains, and what its goals are. An application that has proper usability naturally fills the requirements set by standardizations, and these can be monitored for example with UX and user

requirement testing. Usability testing is done by UX specialists [9]. Often when evaluating usability the heuristic model by Jacob Nielsen in 1994 is used [9]. The ten usability heuristics are the following

- visibility of system status
- match between system and real world
- user control and freedom
- consistency and standards
- error prevention
- recognition rather than recall
- flexibility and efficiency of use
- aesthetic and minimalist design
- well-structured features
- use of default values

[9]

The user experience of a web application can also be tracked differently depending on who is acquiring this information, for example a company might include earnings and page uptime in their metrics for user experience [30]. Some better metrics for a good user experience in general are the "HEART" metrics, that stand for happiness, engagement, adoption, retention and task success [30]. These metrics guide analyze the user's feelings within using the application. The data for the analysis can be acquired for example with surveys or using human-computer interaction, where signals are acquired with for example the camera or with eye tracking. The surveys on the topic can use the Likert scale, where statements are answered with a five- or seven-point system on agreement [31]. Evaluating usability as a part of the UX can utilize the Nielsen heuristics as well [9]. Since UX is ultimately subjective the consensus is, that surveys or questionnaires are currently the best method for evaluating it [32][33].

Cognitive load is a factor that affects the user experience but also usability. It represents the load or stress the user's cognitive system is under when using an application or completing a task [34]. A good user experience enforces that the cognitive load stays small, so that the user is less likely to feel overwhelmed while completing the tasks. High cognitive load also decreases the ability to learn or comprehend new information [34]. The user should use their mental resources to complete the task, not to navigate or understand the user interface or the application itself [34]. Cognitive load is especially important to observe when discussing multimodal systems, that can overwhelm the user more easily than a traditional UI [34]. Therefore UX and cognitive load go hand in hand in UX measurement. Even if the user interface follows best practices and heuristics for good usability, if the cognitive load of the user grows too heavy, the UX will be negative. Respectively a UI prompting good user experience can help reduce the cognitive load of difficult tasks. At worst a heavy cognitive load can increase stress and increase human errors [34]. Higher stress levels increase the overall negative feelings felt by the user, and the whole UX suffers. Therefore mental factors such as stress can also be used as metrics for UX.

In the recent years using artificial intelligence to improve UI designs and usability has increased drastically [35]. Artificial intelligence as a part of the design process for user interfaces has also gained popularity lately. Although the need for UI designers isn't going anywhere there are ways in which AI can be utilized for more efficient design processes. As of now AI cannot be solely trusted upon for designing user interfaces, but a collaboration between a human and AI can help enhance the design process while the critical thinking of the designer still is available [36]. The AI can work as the designers assistant, providing new points of view or identifying usability issues [36]. Especially testing the UI and its usability can be made easier with the help of AI [27].

2.2 Human-Computer Interaction

Human-computer interaction is a field that studies the way humans and computers interact. HCI is important when discussing adaptive interfaces and AI's utilization in the process [37]. It can be viewed as the combination of psychological factors and technology, and this can be beneficial to understand when studying intelligent user interfaces and improving usability with AI. The goals for intelligence in technology are efficiency, effectiveness and naturalness of interaction [35]. HCI is an important field of study especially for getting responsiveness that feels authentic in the interaction. HCI focuses on how to incorporate human feelings, thought processes and behavior into the design of systems. This can be of great value in improving user interfaces to be more intelligent and predictive to the user's needs. There are many different factors that might affect the users needs and wants towards the interface. Some of these factors are gender, age, culture and personality [37]. Some of these factors, such as gender and age, are possible to be observed with different technological devices, and the data could then be utilized in improving the user interface to match the user. Facial recognition plays a role in obtaining this data, even though there still are factors that influence us that aren't visible on camera.

Adaptive user interfaces are application UIs that can adapt to the circumstances. Depending on how the adaptivity is executed in the UI the adaptations can be seen more or less clearly. Adaptive user interfaces can be split to interfaces that have adaptability, and those that are self-adaptive or have adaptivity [38]. The former term describes user interfaces that can be adapted by the user, whereas the latter terms describe a UI where the adaptation happens within the system itself [38]. The combination of both of these is called mixed plasticity [39]. The UI adaptation itself consists of three different aspects, which are the type of information displayed, the level of interaction and source of knowledge [39]. The adaptation can happen based

on a number of factors, that are either set by the user or defined in the system.

In practice user interface adaptation is often done by utilizing deep neural networks to imitate the experiences the user might desire [25]. By using data sets containing information about different users' behavior recorded when using a specific UI the neural networks are able to predict future user behavior. The behavior can be recorded using different tools of behavior analysis, click behavior data and human action recognition [25]. With improved computational power there is a possibility for even more efficient use of real-time adaptivity in user interfaces.

Context-awareness, when discussing adaptive systems, means the systems ability to acquire information about the context in which it is used, and then adapt accordingly even before the application is completely opened [40]. The factors that affect the context information have to do with the environment, the platform and device, the user and the activity [40][41]. For example the used device is part of the context in which the application is used. Some parts of the context are easier for context aware systems to acquire, for example resolution. The systems can also utilize GPS data to gain context, for example in map applications. To extract context about the user, for example their emotional state, other modalities are needed [41]. Other information about the user besides their emotional state could be their age or gender [41]. In a successful adaptive application the application can adapt based on the sum of context information, rules and UI parameters [41]. Acquiring context information can happen using different context models and modalities, and for example facial expression recognition can be used when determining the emotional state [40].

Some popular systems already implement different levels of adaptive user interface design. For example shopping websites or streaming services might be able to give the user suggestions on what content they might want to see next [18]. Although this can be considered an adaptive aspect of the user interface and it might improve

the user experience, it doesn't have an affect on the usability on the website. Some experiments have been conducted on adaptive user interfaces that utilize specifically data about emotional states, but there hasn't been huge breakthroughs to commercially used applications that utilize this data [41][40].

Before the rise of adaptation in UIs there has been personalization. Personalization can be done based on the user's previous choices in certain applications. Since people require different aspects from the same system or interface, there is a need to personalize the interface to match to the user's specific needs. If the UI offers options for personalization the user can improve their personal user experience by making the UI adapt to their own wishes and needs [38]. Sometimes the line between personalization and self-adapting user interfaces can be hazy, especially to the user [38]. Even in literature these terms can be seen used as near synonyms.

Different personality types and different emotions sometimes make the user desire different types of user interfaces, and they get the best experience when the UI matches these factors [37]. Using AI makes the process of connecting the user interface personally to the user's needs more efficient, and therefore the UI itself could be modified faster, improving the user experience. A way to find out how the AI could be best utilized for a specific interface is by creating user personas [14]. The user persona contains information on the main functions the user needs and their personal context [14]. The personas can be used to help understand how different user personas could respond to specific changes in the UI, which then help adjust the AI accordingly to get positive reactions. Tools using artificial intelligence are also beneficial in creating these personas, because the workload might otherwise be heavy [14]. Accurately designed user interfaces that give value to usability aspects help the user feel seen and increase engagement [42].

Although this technique works quite well, it would be even more beneficial for the

user and their user experience if the system itself is able to anticipate the user's needs and make the desired modifications without the user's help or input. The personalization that is often offered also lacks in being able to adapt in real time [43]. If the adaptation could be done by the system, it could also happen in real time. This is an area that can benefit from the use of AI. There are many ways through which AI can be used to improve the usability or the user experience altogether. It can be used for example evaluation, data processing and programming, through which the adaptation of the user interface can be done [38].

Personalization and adaptation can be considered as the pillars of intelligent user interfaces, and therefore they are important factors in improving UX [35]. When AI is used for personalization in the UI to improve the UX it's important that the AI understands the contexts in which it is utilized in [44]. Every-day algorithms on different social media platforms can personalize the users' feeds based on previous interactions, and therefore personalize the content to fit the users' probable preferences [18]. Often these algorithms are designed to increase engagement or monetary profits, but it can improve the user experience too. Adaptivity, in turn, can start as something as simple as responsive web design, where the UI layout adaptation to different screen sizes and gadgets is dynamically adjusted to ensure usability [18]. This can for example mean that the user interface is easy to use on a laptop as well as on a phone. These are also examples where adaptivity is used, but there is no mandatory need for also using AI, although it can make the process more efficient.

Affective computing means computing which rises from the utilization of emotions or is aimed to create emotions [45]. The ultimately aim to recognize, analyze and create emotions [46]. Therefore the acquisition of data on emotional states is crucial for this field of research. Affective computing has been discussed for longer than truly adaptive user interfaces, or intelligent user interfaces have been around. As

early as 1997 Rosalind Picard suggested the definition for affective computing that is still used today [45][46]. Picard stated that systems that could take human emotions into account would be more usable and preferred than those that did not [46]. At the center of affective computing is human-computer interaction that feels natural to the user. Affective computing analyzes emotions, reacts to them and arises them, and machine learning has made emotion detection more tangible [45]. The algorithms that are utilized in affective computing are trained with multi-modal datasets [47].

The two main branches of affective computing are emotion recognition and sentiment analysis [46]. The emotion recognition of affective computing focuses on multi-modal data acquired to detect the emotion [46]. This can be for example speech recognition or facial expression recognition. Sentiment analysis predicts if the emotion is positive, neutral or negative [46]. The more modalities of signals the system has the more accurate the predictions can be, since signals can be combined to a multi-modal fusion [45]. With enough data sets the system is able to provide signals that carry emotional tones to them [45].

Artificial intelligence can be utilized in designing an affective UI to be adaptive in real-time [39]. By modifying the UI to be adapting to the specific needs the usability improves creating an altogether better experience for the user [39]. The interfaces that are able use AI to improve the UX and usability by adapting are called intelligent user interfaces or adaptive user interfaces [48]. The latter term has also been used for describing any user interface that can be optimized to the user in any way, and this includes personalization chosen manually by the user [48]. The adaptiveness can be achieved with different algorithms and AI isn't always necessary in those cases. In recent years the number of user interfaces that are claimed to be intelligent has increased, and the term itself has also gained more popularity [35]. With the rapid development and trending of AI it is just a matter of time when user

interfaces can be improved purely with the help of AI even more, so that besides adapting to the user's current needs it would be possible to anticipate the user's wants and needs successfully beforehand [18]. Multi-modal systems can be of help in achieving this.

Altogether the aim is for the user interface to work as a natural partner to the user. There are challenges in how to get the UI to match the preferences of different people if it is trained using the same deep learning models. Even after the UI is perfected to the user's needs their preferences might change over time, so the UI has to stay adaptive in these cases too [43]. When the UI is truly responsive to the user's emotions the user is easily immersed in it which makes a successful, positive user experience [49].

2.3 Image Recognition

Computer vision means the ability of a computer to read visual input, such as video feed or images [50]. The most utilized applications of computer vision are for example object recognition and image classification, which are also used in image recognition [50]. For computer vision to work there is a need for the correct technology, which is neural networks. Neural networks are especially beneficial for feature extraction, which is essential in image recognition [50].

Convolutional neural networks have multiple layers, and at least one convolution layer [51]. In addition to these the CNNs include for example pooling layers and activation layers [52]. The input for the CNN, such as an image, is converted into pixels with specific values, that can be read by the CNN. The feature extraction needed for image recognition is done in the layers of the network. The first filters catch low-level features, such as edges and color variations [51]. Each following layer captures more and more complicated features, for example shapes [52]. Once all

possible levels of abstraction are captured the network is able to form a prediction. For feature extraction all the features have to first be correctly labeled, which is done with the correct training data set [52]. Similarly parts of facial expression can be considered features that are labeled for the CNN to correctly capture and form the prediction. After the feature extraction for image recognition the image classification has to be done to receive the final output. The output can be for example probabilities, forming the final prediction. CNNs are exceptionally efficient in image classification tasks, and they respond well to small changes [53].

Image recognition systems using CNNs often give the output and prediction with confidence scores. The prediction itself is based on confidence as well. For example Amazon Web Services' facial expression recognition model Rekognition provides the output this way. Before making the final prediction, in the final layer of the CNN classification is done. Here the prediction to be given is chosen based on confidence scores for different options or categories [54]. The classification can be done between two or multiple options. The option with the highest confidence score is the prediction, and the confidence score stands for the probability being the correct solution [54]. The functions on the output layer of the CNN can use different methods and different factors to calculate the confidence score. To get the most accurate confidence score regarding the true, correct prediction calibration should be done [54]. More often than not the given confidence score is uncalibrated, and it consists purely of performance of training and validation datasets [54]. Especially difficult to score correctly are features or categories that are not well represented in the data that was used to train the used model [55]. These situations can result in overconfidence. This issue happens when the model is met with input it isn't familiar with, but based on the training datasets it still gives a high confidence score to a wrong prediction. Multiple different classification models have been suggested to mitigate this issue with confidence scoring, such as ensembles and calibration

with histograms [54][55]. Post-hoc calibration has gained popularity as a normalizer between a possible false high confidence score, since it can be implemented without changing the architecture of the model [56].

Nowadays many providers have their own image recognition models usable with APIs. APIs are known for having a black-box problem, where even though the API can be used successfully for a program or application to work, the source code isn't available [57]. The black-box problem means that the user of the API has no way of seeing everything that happens once the API is called. The same black-box problem has also been discussed regarding AI [58]. In general AI users don't know the specifics behinds the models they are using [58]. Machine learning and deep learning are fields with complex technologies behind them, and the architecture of the models isn't commonly open for the public [58]. Since image recognition models provided by different companies are usually used by APIs the black-box problem is present with image recognition and facial expression recognition as well. Sometimes there is no available information about how the input values are analyzed and how the scores are derived. The datasets on which the models are trained might not be available for all. Therefore it can be harder to examine the correctness of the scores, or evaluate possible biases in scoring.

2.4 Facial Expression Recognition

Facial expression recognition is a technology that utilizes artificial intelligence and its methods as a means for recognizing human emotions [59]. Different combinations of models and technologies should be used to get the FER system to match the specific tasks it's desired to accomplish. The basic emotions that all systems often are designed to recognize are happiness, anger, sadness, surprise, disgust and fear [60]. Different systems may recognize less or more emotions. The ability to recognize

and distinguishing similar emotions from one another differs from system to system. Even well trained FER systems can struggle with recognizing certain emotions, and they can make mistakes [59].

One of the most commonly used set of emotions is the one set by Paul Ekman in 2003 [61]. Ekman argued that the six universal basic emotions can be recognized with facial expression regardless of culture, and the emotions are happiness, anger, fear, disgust, sadness and surprise [61]. Although Ekman's is not the only model for basic emotions it is widely used [61]. Many facial expression recognition models also utilize these basic emotions, or some of them. Many crowd-sourced datasets are also labeled based off Ekman's model [61]. Although Ekman's model has been criticized it still is used as a base for many facial expression systems, because scientific research has also backed his model up [62]. Some of the issues that have been brought up are that cultural differences can sometimes be seen in the display of emotions, and for FER the environment affects the input so much, that this set of emotions might not be sufficient [62].

Facial expression recognition can help make any human-computer interaction feel more meaningful and resemble actual interaction between humans [59]. Appropriate responses to the emotions the user is feeling also increase the user's trust in the program [63]. Since facial expressions are a major part of communication, with the information about the emotions the user is possibly feeling, the computer can adapt its responses accordingly, so that the user gets a more positive experience [59]. Although FER technologies aren't perfect and can't recognize all emotions they give great insight on what might be beneficial for the user at the moment. Besides using it to adapt the user interface in real time it is also a great tool for statistics of the users, if there is a possibility to record the footage. Based on the recorded footage and consequent data the designers of the application or interface get a clear image

of what sparks joy in the users and what is confusing for them.

The way facial expression recognition is done in most of the models is

1. Model receives input (e.g. image)
2. Face detection
3. Image pre-processing
4. Landmark detection
5. Feature extraction
6. Classification
7. Model produces output (e.g. emotion)

Building a facial expression recognition system begins with a large amount of facial expression data that is correctly labeled and of a good quality [59] Data sets can be specified by their content and their environment therefore creating different categories for the sets [60]. The data can also be either dynamic, such as a video, or static, meaning still images [64]. It's important to have the data sets annotated in a careful way to avoid any mistakes sprouting from false annotations or labels [59].

There are multiple databases that are available to use, and many FER models are trained and tested on them [65]. Most of the available databases are categorized to reflect the six basic emotions set by Ekman. The data sets can be split into spontaneous and posed sets [65]. There are more posed data sets available, which is why they are still used more often in training FER models [65]. The issue with this method is that posed facial expressions don't always correspond directly with naturally presented emotions. Some of the recognized databases include CK+, FER-

2013, AffectNet and Ferv39k [65]. Different databases contain datasets of tens of thousands of images or video. The databases have specific resolutions, and can have hundreds of subjects. There are also databases specifically consisting of content of certain nationalities or of certain ages. For example there are databases made specifically for children's emotion recognition [65]. Most of the available databases contain images that are posed, but there are also datasets such as the Ferv39k that contain video material captured to be spontaneous [65].

FER systems require a camera through which they are able to see the user's face where the facial expressions are seen. The system has to distinct the face from the background so only the needed parts of the footage are studied. When processing the camera feed it's important to pay attention to part of the program that get rid of excess background [66]. After the face is detected specific facial features are extracted from the feed and passed onward to the model [64].

The way facial expression recognition is performed mirrors image recognition. The recognition process begins with face detection. The base key to face detection is to find the region of interest, this being the face, so that it can be more closely studied [66]. This step of the process can be executed with different detection algorithms, that are chosen based on the requirements of the situation. One of the best known algorithms is the Viola-Jones algorithm and Haar-like features, which use pixel values and integral images for the detection [59][65]. The methods and algorithms used for dace detection can be categorized under knowledge based, feature based, template matching and appearance based methods [65]. Different methods start recognizing sections of the image differently. Knowledge based methods extract facial features such as nose or mouth, feature based methods can extract structural features such as shapes and texture, appearance based methods detect certain patterns using machine learning, and template matching methods aim to match the input

with predetermined templates [65].

Image processing and image pre-processing focus on extracting the main part of the capture for the system. This means that all non-essential objects, such as the background, are removed from the capture [59]. The footage going forward in the system should only include the information needed for the recognition process. If the image contains too much noise or the studied features are not clear to see the system fails [66]. Image pre-processing also includes adjusting contrast and lighting in the way that makes features stand out for the system [59]. Sometimes gray scale conversion is included in the image pre-processing [66]. This part of the process aims to make the image as clear for the system to perform well as possible. Mostly all relevant information from any image will be unaffected even when the image is being gray scaled. Gray scale images also need less power from the computer [66].

The next step needed for all FER models is feature extraction, as it is for image recognition as well [59]. Feature extraction is a detection method, and it aims to extract the most valuable information from the capture [66]. FER models can be split into static and dynamic methods for feature extraction [65]. Static method focuses on deriving the information from a still image, whereas dynamic method extracts features from a sequence of frames, such a video feed [65]. The algorithms that are used in feature extraction can be classified as either geometric or statistical [59]. Common statistical feature extraction algorithms are local binary pattern, principle component analysis and active shape model [59] [66]. Geometric feature extraction aims to recognize the key facial features, such as the nose, mouth and eyes [59]. Some techniques that are used are for example histogram-oriented gradients and local binary patterns [65]. Different FER models are utilizing different techniques.

Similarly as in image recognition, the final step that directly aims to get the results is classification [59]. Classically the algorithms strive to categorize the facial expression

features, which will result in the prediction of the emotion felt in the capture. The classification for facial expression recognition is categorized for example by emotions. The emotion classification is often achieved by using algorithms like support vector machines, decision trees and k-nearest neighbor [65].

Common issues with FER technologies are for example the changing environments [59]. A change in lighting or position of the camera can have an effect on the results, and this kind of variation is not optimal for systems and applications. Overexposure and shadows can affect how well facial expressions, especially subtle changes in them, are detected by the camera and the model [65]. The models might also recognize emotions differently depending on ethnicity and gender [66]. Since most of the available training and testing data is of posed images and video detecting natural and spontaneous emotions is harder for the models. The prediction often is more accurate if the person using the model poses for the camera as well [65].

Mitigating these issues could be achieved by using diverse databases, that contain images of people of different ethnicity and ages, as well as images from good and bad lighting conditions [65]. FER systems often tend to work better in theory than in real life situations [66]. FER models also require a huge amount of storage and they are often heavy-weight [67]. In order to get FER systems more accessible for everyone the systems should be made more lightweight so that even less powerful computers might be able to utilize the model.

Currently recognizing facial expressions and emotions in different settings is being utilized to some point, but it still hasn't reached its peak. FER and emotion recognition have been used in the medical field, games and customer feedback, but it seems there's still no one clear way on how to most efficiently utilize it [68]. The topic of using FER solely to improve user experience and learning more about how the UX could be improved has also been discussed, but lacks in conducted studies.

It's stated that reviewing and analyzing the UX with questionnaires or integrated questions in the UI can by themselves disturb the user and therefore affect the UX negatively [69]. With the ability to see the user's emotions as they are using the UI the analyzing of their emotions and therefore the UX can be more realistic and efficient [69]. Using facial expressions as a way to affect the UI has just been theorized for a few years. There seems to be a lack of utilizing this technology though, and there's more land to explore about how facial expressions could help navigate a user interface [49]. It's clear that this addition to user interfaces would tremendously improve the human-computer interaction and when working well also the UX [49].

2.5 Facial Expression Recognition from Footage

Facial expression recognition can be done from existing footage but also real-time. Facial expressions and changes in them can happen very quickly in the real world, and usually people are able to detect changes in them instantly. The recognition of different facial expressions is a key part of human communication [70]. The detection process is almost instant in our brains, and when can perceive information constantly from each others faces. This is not the case in facial expression recognition models. Specifically dynamic facial expression recognition (DFER) models are trained to work so that recognition happens fast from dynamic input, such as a video [70]. DFER aims to capture the dynamic changes in facial expressions as another human would notice them in real life, essentially tracking changes in the expressions and analyzing them simultaneously [70]. This has proven to be difficult technology and computing power wise. For example 3D CNNs have been used, but there still are few commercial models that are able to track changing emotions real-time [70].

In practice facial expression recognition is done mostly by taking a frame from the dynamic input, since static facial expression recognition models are currently more

reliable. The frame is then treated as a single, static image, which is easier for all models to analyze, and the analysis is done similarly as image detection is [70]. The frequency with which the frames are captured to be analyzed can be chosen by the user.

Third party and external APIs are known to cause some latency [71]. Since even real-time FER models are usually utilized via APIs there is always latency in the process of facial expression recognition. A cloud based model can give the solution is milliseconds, but the recognition is not truly real-time. Still FER models take a lot of computing power and latency issues increase when resources are scarce [72].

Another issue with API based facial expression recognition is the amount of API calls needed for a real-time, dynamic output. Since APIs need the input from the used device, such as a frame from a video feed, frequent calls mean frequent capture of frames. This can easily lead to a memory issue with the used device [72]. The taken frames can be large in resolution, and therefore might need to be adjusted before calling the API too. The connection to internet also has to be sufficient when cloud APIs are called so that possible connection issues wouldn't cause additional latency [72]. Since using APIs certainly cause some latency always, it isn't optimal for dynamically and continuously recognizing facial expressions. The user won't get a real-time adaptive experience with using FER APIs.

For a more efficient approach to facial expression recognition using a locally run FER model can decrease some of the latency coming from using external APIs. Locally run FER models can access the needed information without using the internet or APIs. A locally run FER model needs immense amounts of memory and computational power, so this isn't a realistic choice for all environments [72]. With a locally run FER model any network issues are also mitigated [73].

Training a custom FER model can mitigate some issues that are present when using cloud based FER APIs. By using a custom trained and custom built FER model the user is in charge of how the predictions are made, i.e. how for example neural networks are designed in the model. By using custom training and testing data sets the user can be certain about correct and incorrect predictions. The use of a self-made, custom FER model eliminates the black-box problem of not knowing what is going on behind the system. However building these systems isn't easy, which can be the reason for the rising utilization and approval of API used FER models.

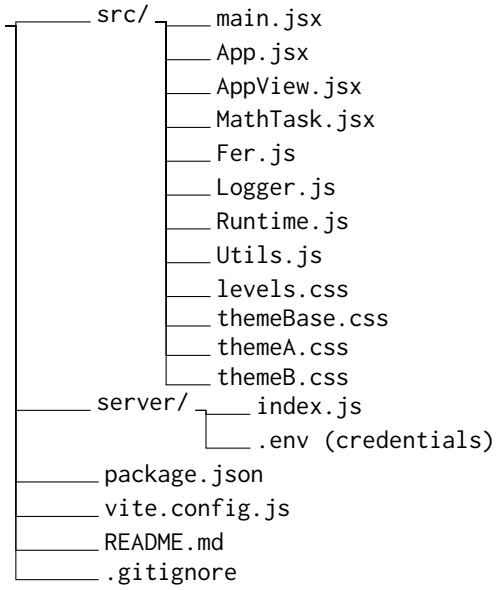
3 System Design for Case Study

This chapter gives insight into the technical attributes of the system that was built for the study. The system was built to be used on a single laptop with one screen and a built-in web camera. Artificial intelligence programming tools provided by ChatGPT 5.3 were used for assistance in programming the application code.

3.1 System Design

The system built for the study was a web-based application with an adaptive user interface. It utilized two facial expression recognition models from two providers. The application was created to be run locally. The frontend was built using React, and the program included three JSX files. Backend was created using Node.js, for framework specifically Express. The backend was needed for passing information between the frontend and the external APIs. API keys were used in the program, which made the backend crucial, to keep the keys safe. The FER systems that were integrated to the design were Rekognition by Amazon Web Services and Google Cloud Vision API. Both the frontend and the backend needed to be running at the same time for the program to work on the laptop. The hierarchy for the built system is seen in Figure 3.1. The full code can be found on `https://github.com/sarakeskilohko/fer-adaptive-ui`.

Figure 3.1: File structure of the system



The frontend handled UI changes and the logic behind it, basic functions necessary for a successful UX and camera capturing. It used React. The captures from the web camera were taken at intervals of 6 seconds, and they were sent to the backend as base64. The intervals for the captures were kept frequent to ensure true adaptation to facial expressions. The captures were requested with `getUserMedia`. From the video feed the frames were extracted using canvas element and transmitted to backend using a HTTP POST-requests. The frames were preprocessed and downscaled to 320 pixel compressed JPEGs before they were sent to be analyzed. This choice was made to keep the needed memory load slightly smaller. The API calls to both providers were simultaneous and done in the same request.

The frontend rendered the single page application, which contained a UI with instructions, exercises and camera settings. Mathematical exercises were implemented solely in the frontend. The rendering for the this was done with the help of KaTeX, developed by Khan Academy. KaTeX is a LaTeX based mathematical documentation tool for web based rendering [74]. Besides KaTeX, rendering was done with

returning HTML for the web application.

The frontend also contained a randomization for choosing the FER engine for each run. Even though the output from both FER engines was logged at the same time the logic for UI adaptation required the results from just one provider. To keep it unbiased a randomizer was created in the frontend, so the application chooses the responsible provider for each beginning run.

The backend used Express as a framework and Dotenv Node package manager (NPM). It included functionality for both AWS Rekognition and Google Vision. Environment variables were necessary for the external APIs, and API credentials were securely stored. The need for a backend also stemmed for safety measures regarding API keys. The FER systems were different to the extent that it was necessary to create two sets of functions. Both AWS Rekognition and Google Vision had their corresponding functions for reading and analyzing the values gotten from the FER APIs. Since both FER systems used for the application were cloud-based the frames were analyzed on the cloud. The APIs returned the results as numerical emotion data which was normalized in the backend, before it could be further utilized. This data was read by the logic functions for UI adaptation and it was saved to the CSV files. During a successful run the application created two CSV files containing different data about the run.

The system architecture can be seen in Figure 3.2 and the data flow of a working system is displayed in Figure 3.3.

3.2 System Logic

The system used React states for its changes in content. The first state that the application had was the state, where `isTesting` wasn't yet set to true. Outside of

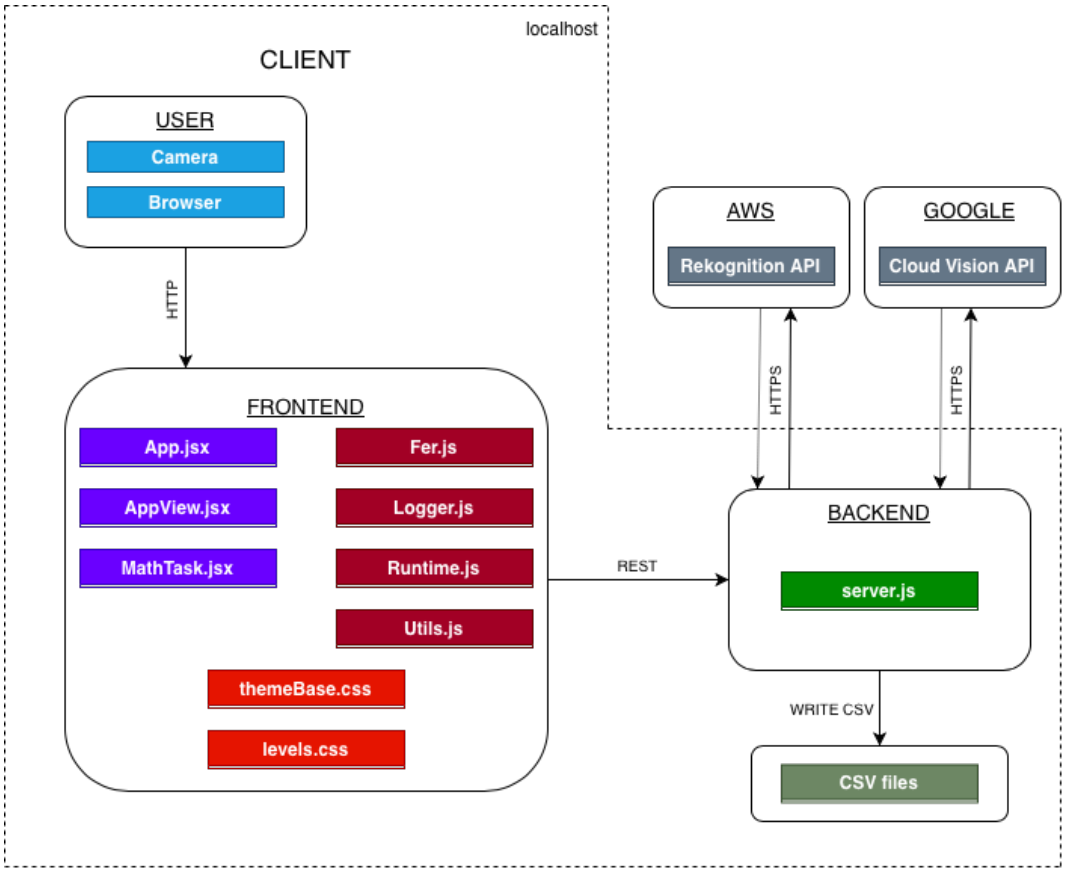


Figure 3.2: System architecture

this state all buttons for manual testing were seen on the UI, and choices between AI could be made. After the camera was turned on and participant code was written the test could be started and this is when `isTesting` was set true. In the testing window the participant wasn't able to see anything that could be used to manipulate the test run.

The states of the application also affected the FER systems' analysis. The FER analysis was turned on when the participant clicked on a button affecting the `isTesting` variable. This way no FER data was received before the participant had read through the instructions and gotten comfortable on being in front of the laptop. The FER systems started getting frames when the actual exercises were available for the participant. Similarly once the final stage of exercises was submit-

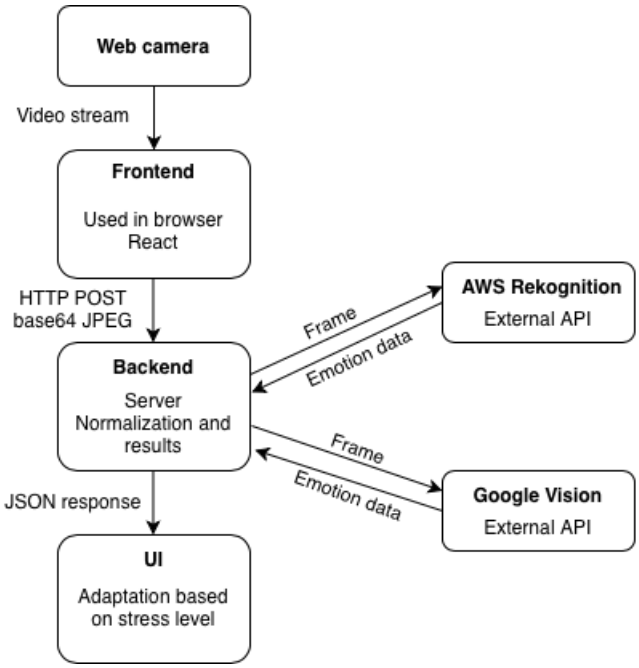


Figure 3.3: Data flow during a system run

ted `isTesting` was changed to false, meaning that the participants filled out the survey while their emotions were no longer tracked.

Since both of the APIs used for the application were able to detect faces it was determined in the program that no UI adaptation or facial expression detection was done while a face wasn't detected. This helped discard analysis rows from the data that contained no useful information. In these cases the emotions detected by the FER systems were consistently classified as "unknown" and therefore couldn't be utilized.

The changes on the UI were based on set numbers. When the distress of the user reached a specified stress level the function enabled a change to a higher support level of UI. The stress level was a value between 0 and 1, and the corresponding changes between levels happened in the following points with the function `stressToLevel()`. These levels are presented in Table 3.1. The UI support level could only increase, so once a more comforting level had been reached it couldn't regress back to the initial

UI state.

Table 3.1: Stress range s to support level

UI Level	**Stress range (s)**	**UI Level**	**Stress range (s)**
0	$0 \leq s < 0.008$	7	$0.056 \leq s < 0.064$
1	$0.008 \leq s < 0.016$	8	$0.064 \leq s < 0.080$
2	$0.016 \leq s < 0.024$	9	$0.080 \leq s < 0.100$
3	$0.024 \leq s < 0.032$	10	$0.100 \leq s < 0.120$
4	$0.032 \leq s < 0.040$	11	$0.120 \leq s < 0.140$
5	$0.040 \leq s < 0.048$	12	$0.140 \leq s$
6	$0.048 \leq s < 0.056$		

The fact that the splits are between values of 0 and 0.2 is determined by the fact that upon testing the application it was noticed that the stress values never reached higher than this point. When a threshold was reached a new UI level was displayed smoothly without changing the content.

The logic for mathematical exercises was a traditional test setting. The UI included a timer, and after the time had run out the indications on correct and wrong answers were displayed, before the participant could move on to the next stage. The math exercises were split into five stages all of them containing five exercises. The UI wasn't adaptable before starting the math exercises, which changed the `mathStarted` variable to true, which in enabled the adapting UI. Each of the five stages was its own state, which were logged in the CSV. The states involved information about why the stage was ended, either submitted by the user or by a timer that was run out, and how many correct answers the participant was able to score. The mathematical exercises in the experiment were hard-coded, so that every participant could have an identical test to perform. The exercises were created carefully and designed so

that each stage increases in difficulty. The last stages specifically were designed to cause frustration even in mathematically skilled participants.

The application was set to have a cooldown window after the UI level was changed. The cooldown was 7 seconds, and during this time the UI couldn't adapt again, no matter what values the FER systems retrieved from the frames. This choice was made not to make the test setting feel too hectic or draw attention away from the tasks at hand.

3.3 Facial Expression Recognition Integration

The frames for the FER systems were captured by the laptop's web camera used in the experiment. The video was streaming constantly throughout the experiment, and the frames were captured from the feed. The material was captured with `getUserMedia()` without audio. The video material itself nor the frame captures from it weren't saved anywhere locally to be accessed after the experiment run had ended. Before the frames were sent to be analyzed they were encoded to Base64.

The frames were polled every 6 seconds in the frontend, and this was done simultaneously with AWS Rekognition and Google Vision API. The first call is done immediately after the participant has started the exercises. A faster rate of analysis could've given even more accurate results, but for the purposes of the study a capture every 6 seconds was justified. The UI wasn't designed to change too fast not to confuse the participant, so there was no need for more frequent FER analysis loops. Between the 6 seconds all variables stayed constant. The frames were then sent to backend, where analysis is done when frames are received. The backend also has a limit for 4,5 seconds between analyses as a safety precaution, so that frames cannot be analyzed too often. The AWS Rekognition and Google Vision were utilized with `app.post("api/emotion/frame")`. Both APIs returned values that they acquired

from the analysis of the frame. The analysis was done in the backend since both of the APIs required security credentials to work. This architecture was chosen to keep the API keys secure.

Before the returned values were used for changing the UI of the application the values were normalized and smoothed. The data from both APIs is normalized so that all values were from 0 to 1. The normalization was needed since the APIs produce values differently. After the normalization stress and confidence were calculated. The stress was explicitly used for the UI changes. The variable stress included negative feelings, and it was calculated with `emotionsToStress()`. For AWS Rekognition the calculation was done with the weighted sum of `0.40 * angry + 0.30 * sad + 0.20 * confused + 0.10 * surprised`. For Google Vision the stress was created with `0.45 * anger + 0.35 * sorrow + 0.20 * surprise`. This together created the stress variable, the values of which were studied in the frontend and used for UI adaptation.

The stress variable was then passed to the frontend. Here exponential moving average (EMA) smoothing was done. The EMA smoothing was utilized so that the changes in stress levels would increase without huge jumps so that the stress-based UI adaptation would happen stage by stage. The smoothing was applied every 0,5 seconds. The smoothing followed the common smoothing formula where the value is multiplied by the smoothing factor `smoothed = previous +` α `(current - previous`. The α used for the experiment was set to 0,25. This means that the stress value moved only 25% towards the actual value of the stress predicted by the FER systems. The newer value was therefore closer to the previous value, and no jumps happened. Finally the smoothed value was read and the UI is adjusted accordingly.

3.4 User Interface

The user interface for the application was designed so that adaptive changes happen level by level, and the jump between the levels wasn't noticeable. The content of the UI was the same for all participants, the differences in experiment runs came from the support level CSS that adapted based on facial expressions. The aim of the changes in the UI weren't so that the focus on the tasks would falter, but rather to ease some stress that calculations with time limits were inducing.

The user interface included a header, containing a description of the study and instructions for the participants. A navigation bar was implemented for navigating the application. This included all necessary buttons for controlling the camera, setting the participant code. Before starting the experiment the UI displayed more buttons, and once the experiment was started, all buttons not necessary for the participant were hidden. Finally a panel for math exercises was added to the interface. This panel is where the five sets of five exercises were displayed for the participants using KaTeX. The layout stayed the same throughout the whole experiment, to keep the focus on the exercises, not on navigating the UI.

The UI had a base CSS that is applied always. In addition there were rules for each of the 12 UI support levels, that separated the levels from one another. The main differences between levels were the color schemes, font sizes and shapes. The user interface was defined so that for lower levels the color scheme were dark blue, and the more distressed the participant was the more the support level increases. The highest support UI levels included more bright colors, such as warm yellow. The shape and size of the buttons on the UI also changed gradually with each support level. From level 8 onward there were small icons floating in the header of the screen, designed to lessen the stress felt by the participant. The icons were colored to match the UI's current color scheme, and they were only placed on the top of the screen

not to disturb the exercises themselves. Figure 3.4 visualizes the header on a high support level.

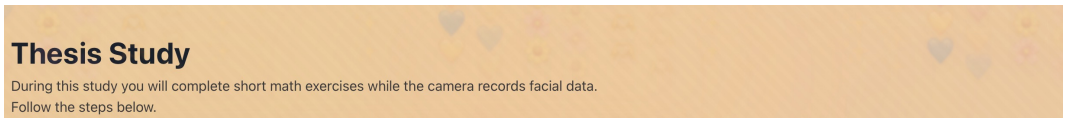


Figure 3.4: Icons in application header at support level 10

Once a specific support level was reached based on the logic for UI adaptation the correct CSS was set to be enabled. Continuous smoothing was applied in nearly all elements of the UI to keep the changes between UI support levels discrete and pleasing. With even more support levels the changes between each support level UI could've been made even more seamless. Although the UI had modest animations for this reason, all animations were disabled in the math exercise panel. This was done to avoid any additional lagging that could be avoided easily.

The color schemes of the application and its different UI levels are shown in Figure 3.5. While the increase of colors was designed to increase comfort and decrease stress the colors weren't as bright as they could've been. The point of the different colors wasn't to be distracting, and bright neon colors might've lessened the engagement of the participant. The choice for using shades of yellows and oranges was made to prompt positive emotions, since these colors have been linked to emotions of warmth and joy [75].

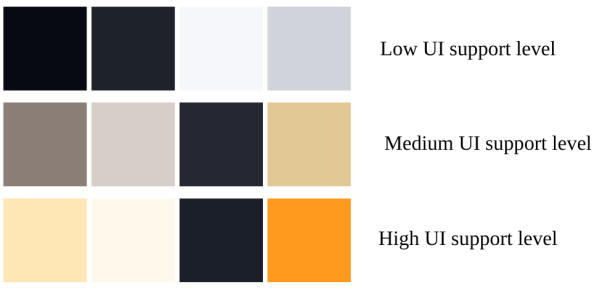
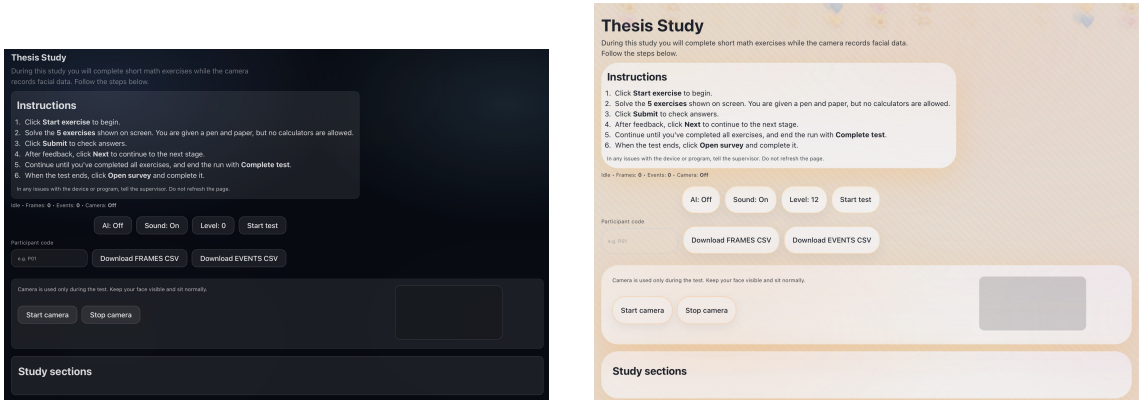


Figure 3.5: Color schemes in the application

The user experience was in the core of the program, and therefore the UI was designed so that the user essentially pays no mind to how the interface looks. The changes in emotions prompted by the UI changes could happen subconsciously. For easy navigation buttons were at points highlighted to guide the user to proceed. This was done with `.nextAction` in the CSS. Once a certain support level was reached it was not lowered back to a previous one during any point of the test, since this jump could've caused more negative emotions. Figure 3.6 shows the whole UI with the supervisor view on support levels 0 and 12.



(a) UI support level 0

(b) UI support level 12

Figure 3.6: The whole user interface of the application

3.5 Data Acquisition and Logging

Data logging was done in the frontend and backend. The backend logged the numerical values that were returned by the FER APIs, and the frontend logged information about the UI level that was displayed, as well as the progress of the math exercises.

The application was set to produce two different CSV files. One file contained information about the frames captured by the program and the analysis that was conducted by the FER engines. The event CSV contained information about set events in the application. These events were proceeding in the math exercises.

This CSV also calculates the time that was used in performing the exercises. The information was used to reflect on stress levels, and whether stress levels were lower when the participants were able to perform all tasks within the time limit.

The information about recognized emotions and the UI level provided by the application are logged in a CSV containing the rows displayed in Table 3.2.

Table 3.2: Structure of the Frame CSV File

Column Name	**Type**	**Description**	**Purpose**
participantCode	String	Participant identifier	Data grouping per participant
testId	String	Session identifier	Distinguishes separate runs if many per participant
timestampISO	ISO datetime	Time of frame analysis	Timing of different exercise stages
elapsedMs	Integer	Milliseconds since starting exercises	Analysis for how long stage completion took
phase	String	Which math stage is being completed	Analysis for emotions per math stage
aiEnabled	Boolean	Are FER engines used for analysis	Verifying that FERs are working
smoothedStress	Float	Stress value derived from FER system in use	Validating UI changes based on stress

Column Name	Type	Description	Purpose
supportLevel	Integer	UI design level	Analysis on effects of each level
uiDriverProvider	String	Which FER system was chosen for UI adaptation	Analysis of two different providers accuracy
usedStress	Float	Stress value that was used for adaptation	How stressed participant felt at different moments
usedConfidence	Float	How certain is the emotion prediction	Analysis for FER engine performance

In addition to these for both AWS Rekognition and Google Vision API specific information about the run was logged for the analysis of the performance of these FER engines and their accuracy. The logged information is displayed in Table 3.3.

Table 3.3: Logged data for FER engine performance analysis

Column Name	Type	Description
Stress	Float	Stress level from the FER engine
Confidence	Float	How certain is emotion predicted by the FER engine
RawEmotions	JSON object	Numerical data values for each recognizable facial expression

Column Name	Type	Description
AiSkipped	Boolean	Was the specific frame used for UI adaptation
Faces	Integer	Does the FER engine in question recognize a face in the frame

The information about succeeding in the mathematical exercises and the duration that completing these tasks took is logged in the CSV for events, which includes the rows displayed in Table 3.4 as well `participantCode`, `testId`, `timestampISO`, `uiDriverProvider`, `phase`, `supportLevel` and `elapsedMS` which are also seen on the CSV dedicated for captured frames.

Table 3.4: Structure of the Event CSV File

Column Name	Type	Description	Purpose
stage	Integer	Math exercise stage under completion	Performance segmentation
answeredCount	Integer	Number of submitted answers	Analysis of performance in relation to stress and task difficulty
correctCount	Integer	Number of correct answers	Accuracy measurement per stage
durationMs	Integer	Time required to complete the stage	Task duration analysis

All logged information was used either to study how the UI changes affect the emotions felt by the participant, or how well what the participant felt correlated with what the FER engines predicted for the participant's emotions based on facial expressions. Participant codes and test ids were needed, because the survey for acquiring data about the participants' individual experience wasn't integrated in the application UI. When the survey was opened after the experiment was run successfully the participant code was automatically set to the correct one on the survey.

3.6 Amazon Rekognition

Rekognition by Amazon Web Services is a facial expression recognition system that is developed by Amazon Web Services. It's cloud-based, and can be used for both pictures and videos [76]. It has multiple uses besides using it for facial analysis or facial detection. The main use cases for the product are face-based user identity recognition, label and object recognition especially regarding media libraries, face liveness detection and facial search [76].

Face detection in Rekognition is done by analyzing facial landmarks [76]. Once a face is detected, the landmarks are analyzed, and based on this the values and confidence scores are given.

AWS Rekognition can give out nine different values based on what the FER model can recognize from the frame. The values are `HAPPY`, `SAD`, `ANGRY`, `CONFUSED`, `DISGUSTED`, `SURPRISED`, `CALM`, `UNKNOWN` and `FEAR`. These correspond with Ekman's basic emotions. The values that are returned by the API are floats, such as 82.6. In the experiment application all of these values acquired were normalized in the program code to be percentage values, such as 0.826. Out of the possible emotions, anger, confusion, sadness and surprise were the ones that factored in the stress value,

and the weights for these emotions in the stress factor were calculated with `0.40 * angry + 0.30 * sad + 0.20 * confused + 0.10 * surprised;`. For confidence on the emotion recognized Rekognition provides a value between 0 and 100, ultimately a confidence percentage [76]. An example of how the emotion predictions are provided is seen in Figure 3.7.

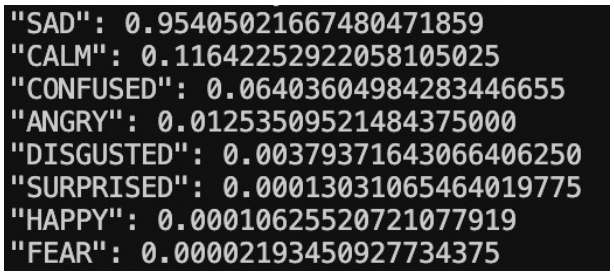


Figure 3.7: AWS Rekognition output

3.7 Google Vision

Google Vision, or sometimes called simply Vision API or Cloud Vision API, is a cloud based API, that can be used for facial expression recognition. Its more popular use cases, however, are for example text and landmark recognition [77]. The input for Google Vision has to be given as base64 encoded images [77]. Besides face detection and facial expression recognition Google Vision can be used for custom purposes and labeling custom content is possible with Google Vision [77].

The Vision API returns likelihoods of emotions, rather than the emotion confidence scores themselves. The API has six distinct values it can return, which are `UNKNOWN`, `VERY_UNLIKELY`, `UNLIKELY`, `POSSIBLE`, `LIKELY`, or `VERY_LIKELY` [77]. The vagueness of these return values indicate, that this FER API only works for specific cases. Google also informs that specific individual facial recognition isn't implemented in their API [77]. The emotions which can be recognized and used for the likelihood predictions are joy, sorrow, anger, surprise [77]. Google Vision doesn't have a "neu-

tral" or "calm" emotion to recognize, unlike most FER models. The five stages of likeness work similarly as AWS Rekognition's confidence, although since its not a continuous scale the values aren't as descriptive. The values that the likelihoods are bound to can be set by the user, and in the application the corresponding numerical values can bee seen from Table 3.5.

Table 3.5: Likelihood scale for Google Vision

Label	**Value**
VERY_UNLIKELY	0.00
UNLIKELY	0.25
POSSIBLE	0.50
LIKELY	0.75
VERY_LIKELY	1.00

Based on the likelihoods the stress score for the application was calculated with using emotion likelihoods from sorrow, anger and surprise with weights `0.45 * anger + 0.35 * sorrow + 0.20 * surprise`.

Google Cloud Vision API recognizes faces using bounding polygons, and the facial expression recognition is completed by the engine's studying of landmarks [77]. The landmarks are for example eyes and nose.

3.8 Constraints in the Technology

Due to the budget limit for the thesis the API calls were restricted to a set amount per day. This restriction was done by environment variables and the backend. Since the application was run on an older laptop with little memory it had to be designed to be still manageable. This means that there was latency.

Constraints that were posed by the providers of the technology were the range of emotions recognized, and the way the recognition was returned as values. The two FER model APIs also worked differently, which prompted the need to normalize results gotten from the APIs so that they would be comparable. The most significant differences are displayed in Table 3.6.

Table 3.6: Comparison of AWS Rekognition and Google Vision

Criteria	**AWS Rekognition**	**Google Vision**
Recognizable emotions	8	4
Output	Continuous float (0-100)	Categorized (5 points)
Confidence	Confidence/probability score (%)	Likelihood of emotion
Stress calculation	angry, sad, confused, surprised	anger, sorrow, surprise

AWS Rekognition's strength was that it was able to recognize many emotions, and the most common emotions were on this list. Rekognition also returned a confidence score, that was calculated into the predicted emotions by the program. Google Vision however was only able to return likelihood values, and these likelihoods were regarding only four emotions, which were joy, sorrow, surprise and anger. This was a very narrow list of emotions, and a successful integration of Google Vision was more difficult than Rekognition for this exact reason. With likelihood values of only four emotions the API often returned "unknown", meaning that no emotions from their list of allowed emotions was detected. The likelihoods weren't returned as continuous numbers, but as discrete steps. Therefore changing the steps between the values didn't give any additional information; the steps were still the five likelihoods.

Neither of the APIs included ways to train own datasets for the detection of more emotions. For example Google Vision could've been improved with the possibility to recognize a multitude of emotions. Practically the program had to built around

what the FER systems were able to provide, and modifying the systems to better correspond the target system wasn't an option.

Both APIs needed to be used on the cloud. This meant that for the application to work a stable internet connection was required. When discussing large amounts of data moved between the client side device and the cloud this can cause latency. The fact that both APIs were cloud based also meant that the facial expression recognition could not be purely in real-time with what the camera captured, since a little latency is always included in cloud based systems.

The obvious constrictions for both of the APIs are camera-related issues. Poor lighting, bad resolution or wrongly setup camera can ruin the whole experiment. In the experiment done for this thesis it was ensured before each run that the lighting was sufficient and that the camera was set up so that the participant's face can be clearly seen. Even though these constrictions weren't met in this experiment they are still significant issues that can arise in a similar study. For long time usage both of the FER models can also become quite expensive, which is why budget restrictions were set to the application.

Both FER APIs include the black-box problem present often with APIs. Neither FER model indicates what databases or datasets were used in the training, and there is no visibility on how the APIs work in practice on the cloud. The only output the user sees are the confidence scores the APIs provide. How confidence scores are calculated specifically is also in the black-box problem for both FER models.

As for all facial expression recognition models the systems are only able to recognize what is shown on the face. Some participants might have not been expressive, in which case the emotions aren't logged either. This is an obvious constraint with all FER systems, especially when the system is not multi modal.

3.9 Ethical Aspects

Ethical issues regarding AI and FER have been discussed for years. This thesis isn't going to go into detail about all ethical aspects of using AI, but rather mentions the most important ethics to consider when using facial expression recognition. Although AI regulations are more and more discussed and EU has had some bills about stricter laws regarding the use of AI, as of 2026 the AI field isn't properly regulated. EU has implemented the AI act which regulates some areas of AI use and touches facial expression recognition too in a way, for example prohibiting compiling FER databases from sources such as CCTV [78].

When discussing AI in facial expression recognition it's important to pay attention to the privacy of everyone on camera. Cameras can utilize face detection or facial expression recognition sometimes without the people on the camera knowing about it, which raises concerns [79]. People have the right to know how their data is processed, so this is an issue with consent [79]. With privacy concerns it's also important to note that the providers of these models and applications have to care for their privacy and security, since they process individual data that can be used for identifying [79].

Biometric data, specifically facial data is a form of information that is being used more and more by companies all over the world [80]. This makes it crucial to consider the ethical concerns on utilizing facial data, since some companies are known to utilize the facial data even without consent [80]. Especially facial recognition, through which an individual can be recognized, poses serious ethical issues. Facial data can be used for detecting an individual in public even when they aren't aware their facial data is being utilized, and this can lead to major problems such as discrimination and manipulation [80].

The ways that facial data is used in training AI based facial recognition models isn't always ethical either. Companies can perform web scraping, using images on the internet for training their model, or even collect it with placing cameras in public spaces [80]. In European Union the General Data Protection Regulation, GDPR, lists facial data as sensitive biometric data in Article 9, and this creates strict restrictions for how facial data is processed and stored [80]. The GDPR doesn't explicitly list emotions as identifying data, and as of 2026 there is less research done on emotion recognition and its ethical issues than on facial recognition and its issues.

As with all cloud-based APIs, using FER models through APIs have security risks. Cloud-based APIs can be vulnerable for hacking and malicious API traffic, and therefore need to be designed to be secure [81]. The same risks can affect any FER APIs, and although the ones used in the experiment were by internationally known providers it doesn't mitigate all risks. In FER APIs the information sent with the API call is biometric data, and therefore vulnerable. Using a third-party provider for APIs and cloud services can increase risks for information hijacking or leaking and exploitation [82][83]. Still cloud computing includes a variety of security measures that are and should be implemented to keep transmitted and stored information as safe as possible [82].

Another issue that comes with cloud-based solutions are the geographical matters. Popular cloud services more often than not are located in multiple different countries and even continents [83]. This can reduce the individuals' control over their data, since the physical location of the data isn't known [83]. With the use of cloud-based APIs for facial expression recognition, as with the use of all cloud services, the user has to put their trust in the data processor and the cloud service provider. A way to mitigate the risks that come from using cloud services would be to process all the facial data locally, therefore avoiding using the internet and use of remote services.

As it has been stated previously, FER models can sometimes be biased. The data that the models are trained on can lead to outcomes that are discriminatory, which can in turn lead to more dire issues such as racial profiling [79]. It has also been discussed whether the basic emotions that most FER models recognize are sufficient, with people from different backgrounds presenting their emotions differently [84]. Another issue with FER models is the previously mentioned black-box problem, especially with APIs. An every-day user only knows what the company wants to share about their system or model, leaving users essentially in the dark.

Ethical facial expression recognition has established best practices that should be followed for as ethical use as possible. These practices include for example informed and adaptive consent, transparency, cultural sensitivity and privacy and data minimization [85]. Algorithmic bias has to be mitigated as well as possible, and participants shouldn't feel uncomfortable while they're being recorded [85]. It is also stated that commercialization should be regulated, so that companies wouldn't be able to provide FER technologies purely for money without any regard for ethics [85]. Needless to say there is still a long way to go to have FER as a working part of the daily life, but steps have been also taken in the right direction.

As for the experiment done for the thesis, before the experiment the participants were made aware that the camera would be on but no content will be recorded to be used later on. The data in the application was only saved locally. The locally saved files were two CSV files per run, containing data on predicted emotions, UI support levels and math stage progress and accuracy. These files only included a test run number generated by the program and a participant id set by the experiment supervisor. No data such as names, ages or genders are saved or analyzed for the research.

For the study only facial expression recognition was utilized, and no facial recognition was done. This ultimately means that only facial landmarks were being detected, and although this is still biometric, personifying data the APIs that were used were not able to perform pure facial detection, for example detecting the same person in front of the camera multiple times. An question that can rise from utilizing facial expression recognition this way, is whether the participants were fully displaying the emotions they felt. The camera could only capture what the participants were comfortable expressing, meaning that the application ultimately was only able to recognize what the participants were comfortable expressing.

The frames from the camera are handled in the backend to keep them secure, and sent to both APIs. AWS Rekognition security is built on top of AWS Identity and Access Management (IAM), which ensures secure usage for clients [76]. In the experiment the only access was for the experiment supervisor, and no one else was able to access the traffic in the API. Amazon also advises best practices for privacy in all their products. They handle their privacy policy according to GDPR, and since all parts of the experiment were done in Finland, GDPR applies [86]. Likewise Google Cloud has vast privacy policies regarding all their cloud based product [87]. They too base their policies on GDPR [88]. Both providers state that the data handled by APIs is not being used for further training of FER models [76][87].

If a system like the one built for this experiment was in commercial use more ethical aspects would have to be taken into account. For a study as small as this one it was quite simple to gain the consent for utilizing cloud-based APIs for processing facial expressions recognized with a camera, and the participants were made aware that their data wasn't being stored locally or in the cloud. The ethical issues rise if companies or websites started utilizing similarly behaving applications without the user being aware of this happening. If the data collection and analysis was

happening without the user's knowledge the data could be used for profiling and manipulating experiences. This stands to show how important it is to follow the GDPR, since it requires transparency if data is being recorded and stored [89].

4 Empirical Study

The case study investigated the abilities of two facial expression recognition models, Amazon Rekognition and Google Vision, on how well they perform in accurately analyzing the facial expressions. It was also used to evaluate whether an adaptive UI could affect the user experience positively. The study was constructed so that the adaptiveness of the UI was determined based on the readings of the two FER engines.

4.1 Research Design

The application examined for this thesis was a web-based application with an adaptive user interface. The user interface was adapted based on the results from Amazon Rekognition and Google Vision. The study involved 25 people of different ages and backgrounds. The participants were chosen based on willingness to participate and convenience. The participants received no compensations for their participation. The study was experimental, and was done in a controlled setting.

The participants completed the experiment individually, and the setting was the same for all participants. For each participant the application randomized which provided or FER model would be the one to drive the adaptation of the UI. The randomization was done so that each participant could only complete the tasks once,

as not to be prepared for another round of similar tasks if they had to complete the test twice with the different providers. The experiment required a well-lit room and a laptop with an internet connection. The participants were also given a pen and one piece of paper. This was done so that the mental resources would be directed to solving the questions, not stressing about not being able to use simple mathematical methods for calculations. Everything in the UI content stayed the same despite the FER provider. The participants were instructed to follow the directions on the user interface. These directions explained to the participants how they were to complete the tasks at hand for the experiment. After completing the tasks the participants were directed to a survey where they were asked to answer questions regarding the experiment, and how they felt during it.

The mathematical exercises were chosen so that the difficulty increases with each level. The levels were designed so that it would increase the participant's stress levels while the calculations still remained achievable at first. The final math stages were designed to be so hard that every participant could feel some stress. If the stress levels of the participant increased according to the used FER systems the UI adapted to prompt more calming emotions in the participant. The FER systems monitored the camera frames frequently so that the UI could adapt accordingly almost in real-time.

The survey questions were conducted based on the research questions of the thesis. The aim of the survey was to find out what emotions the participants truly felt during the experiment, and did the adapting UI affect their emotions in any way. The survey was split into two sections discussing these topics. The FER system for each run was randomized in the application, therefore creating the possibility for comparing two different FER systems and their abilities. The survey and the facial expression data were used to evaluate the FER systems' performance, and

the survey with the UI data were used to evaluate the effects of an adaptive user interface.

The data collected from the study consisted of two CSV-files containing data about UI support levels and the emotions detected by the FER systems, as well as a study where the participants could more accurately explain their thoughts and feelings throughout the experiment. Data about each participant's performance was also collected, to see whether answering correctly decreased stress levels.

4.2 Procedure

All of the experiments in this thesis were conducted with a MacBook Pro 2020 and its own web camera. Each participant completed the test using the same equipment. All participants were made aware that the program uses the camera feed for taking frames, but it doesn't save live video anywhere. The participants were given no information about what the actual study setting included prior to starting the experiment. This choice was made to avoid preparation for the study, for example revising mathematical skills. The participants were given the laptop ready to use, and the study servers were already running. The only instructions given by the experiment supervisor were to follow the instructions on the screen, but if the program suffered an unforeseeable issue, such as a crash, then alert the supervisor of the experiment for help.

The experiment was always started by the supervisor, so that the settings were correct for use. This meant turning on AI, inputting the participant code and turning on the camera. At this point the experiment supervisor could also check the camera lighting with the camera feed preview showing, as to ensure that the lighting conditions were sufficient for the experiment to take place. Then the supervisor started the test run so that information meant only for the supervisor was hidden

from the UI. Once the participant view of the application was enabled there was no possibilities for manually changing the FER model provider or to turn the camera off. This could only be done by aborting the test run or finishing the complete test run. After this the laptop was given to the participant and the supervisor no longer interacted with the participant. The participants nor the supervisor knew which provider was in charge of the UI adaptation during each run.

The instructions on the user interface informed the participant, that they were to complete five sets of mathematical exercises. The exact nature of the exercises were not revealed at this point. The instructions advised the participant to click "Start Exercise" once they felt they were ready to begin. Before the user clicked this no frames were captured with the camera, although the user was able to see the camera feed preview. Under the camera preview it was stated that the camera is only used during the test. This was set up so that the user could check that their face was clearly seen on the feed, as the study required this. The instructions stayed visible on the screen for the whole duration of the test, and the way they were presented is seen in Figure 4.1.

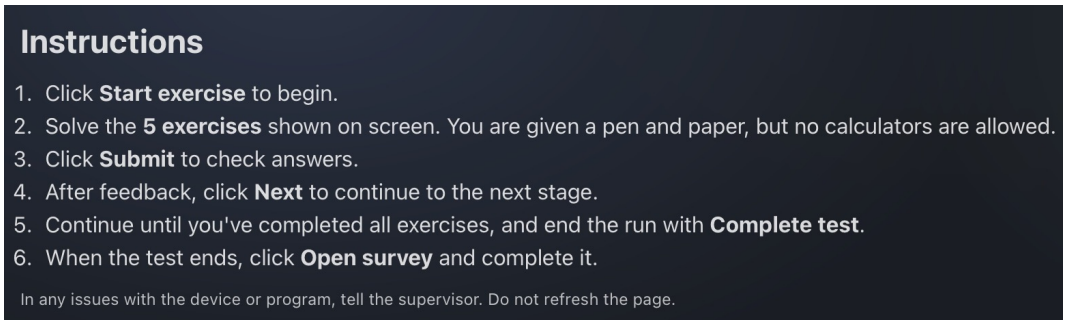


Figure 4.1: Instructions for the participants

After the participant started the exercise the camera feed preview was hidden, so that the focus would be turned to the first stage of exercises. The five exercises in the first stage were additions and subtractions. This stage had a timer of 90 seconds. After completing the stage the user would submit their answers, and the

user interface indicated with red and green colors which solutions were correct and which not. If the timer ran out before the user had submitted the solutions the indication for correct answers appeared automatically once time time was out. Once the "submit" button was clicked the timer also stopped. After this the user could click "Next Stage" to move on to the following set of exercises. The second stage of exercises included multiplication, fractions and one square root calculation. The time limit for this stage was 120 seconds. Moving on to the following stage worked similarly as previously explained. The third stage included simple differentials, specifically derivatives, and the time limit was set to 120 seconds. The fourth stage entailed more difficult differentials, and the time limit was 180 seconds. The final stage had a timer of 180 seconds as well, and the exercises included calculations with complex numbers, one integral, and two approximation exercises. An example of how the exercise screen was seen by the participants is shown in Figure 4.2, and the stage information in Table 4.1. All of exercises can be found in the appendices.

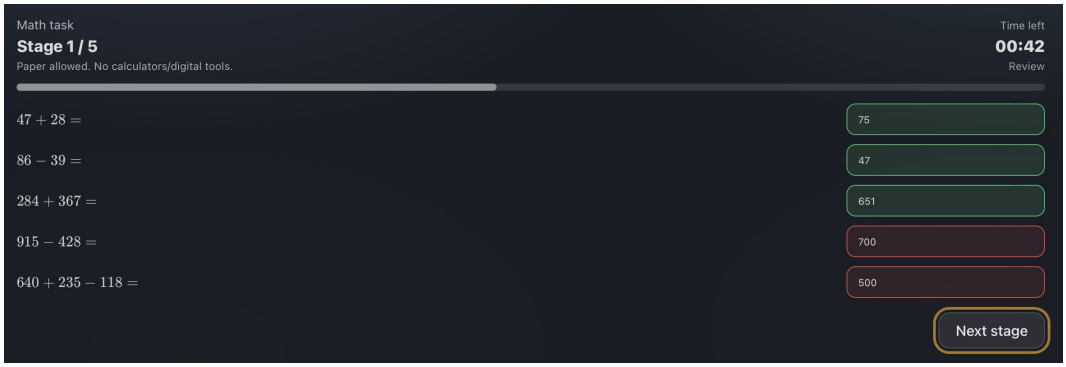


Figure 4.2: Exercise section once solutions were submitted

The stages were designed so that the difficulty increased distinctly after each stage. The timers were implemented on all levels to keep the participants focused on the task, while still creating slight time pressure. Once a stage was submitted by the user or the timer ran out there was no possibility to retake that stage of exercises.

After the participant had completed all five stages of exercises the user interface

Table 4.1: Math Task Stages

Stage	**Difficulty**	**Question Type**	**Time Limit (s)**	**Examples**
1	Very easy	Addition, subtraction	90	$47 + 28$
2	Easy	Multiplication, fractions	120	$125 \cdot 24$
3	Medium	Simple differentials	120	$f(x) = 4x^3 - 7x + 2$. Enter $f'(x)$
4	Hard	Difficult differentials	180	$f(x) = sin(x) \cdot e^x$. Enter $f'(x)$
5	Very hard	Approximation, complex numbers	180	Multiply $(3 - 4i)(-2 + 5i)$

indicated the user to click on an "Open Survey" button, which opened another tab in the same window. The participants then filled out the survey, after which the experiment was finished, and the participants were thanked for their time. The UI adaptation was done during the mathematical exercises. Once the user had submitted the final stage of exercises the interface stopped adapting, since the user would only interact with the survey from that point on. One run for a participant took 10-25 minutes. In order to get the most authentic results the test was only run once per participant, and the randomization of choosing the FER engine for the run was done by the application, not by the participant or the experiment supervisor.

The survey questions reflected the research questions of the thesis. They handled bias from the participant regarding their emotions, emotions prompted by the UI adaptation and emotions prompted by the math tasks. At the end of the survey the

participants were thanked for their time and the supervisor was able to download the CSV files from the run.

4.3 Data Analysis

The data acquired for the data analysis came from the system itself, specifically the CSV files with FER outputs and performance data, but also from the self-reported data acquired by the survey at the end of the experiment. The survey was split into three sections. One section focused on how completing the exercise stages felt, the second focused on how the participant felt overall throughout the study, and the final section aimed to get insight on whether or not the changes in the UI prompted any feelings in the participant.

The first section of the survey was used to determine what kind of stress levels the FER systems should have detected. This section consisted of questions about each stage

- How stressed did the participant feel during the stage
- How the participant would rate the difficulty of the stage
- How confident the participant felt during the stage

In addition to these the participant was asked to rate their math skills to evaluate, if deeper previous knowledge on the topic correlated with feeling less stressed. Hypothetically the more stressed the participant reported feeling during each stage the higher the detected stress should've been.

The second section of survey questions focused on asking what kind of feelings the participants felt throughout the experiment. This section is used for evaluating the overall performance of the FER systems. Since both FER models output raw data

on the facial expressions they detected they can be compared to the emotions the participants reported.

The final section of the survey questions was used to determine whether an adaptive UI has any effect on the emotions felt by the participant. This was determined by studying if an increase in the UI support level resulted in more positive feelings later on in the study. The final two questions of the survey are open questions asking what affected the whole experiment most in a positive way and in a negative way. This was included in the survey to drive the analysis to a correct direction.

After the data was acquired by the application and by the survey it was exported and stored into a folder. The folder was accessed with a Python program designed to build graphs and analyze the data. Data mapping and analysis was done implementing simple Python scripts. The survey answers were exported as a CSV file, and the numerical data from the survey was analyzed. Before data analysis any unusable frames were deleted from the CSV files. These frames were the ones were for example no face was detected, and they could be seen from the files as rows stating `aiSkipped = true`.

The performance of the FER models was done by evaluating the two models against each other. For each frame on each run the calculated stress scores of each provider's model were analyzed to see if the scores corresponded. Stress scores were also analyzed with the progress in the math exercises. The scores were also matched with how stressful the participants reported feeling in the survey. It was also studied how the UI support level changes affected the stress scores. The differences between support level changes between the two providers was also analyzed. The performance of the participants, specifically the number of correct answers per math stage, were also compared with the stress scores, to see if doing well on the exercises lessened the detected stress or increased their confidence in themselves.

The accuracy of both of the FER models were analyzed with the help of participants' reported emotions during the experiment. The raw data of the emotions were compared with what the reportings stated. Free text questions on the survey were analyzed if the data showed abnormalities. The free text questions in the survey inquired what caused the most positive feelings and negative feelings in the participants during the experiment. This could be used to understand possible deviations from hypothesis. It was also meaningful to analyze the free text answers to discover if the exercises were too hard for participants, causing them to give up on the tasks altogether.

4.4 Limitations

The case study has some limitations to its realization. The environment is aimed to be as simple and tranquil as possible on account of outside disturbances that might cause the emotions seen on the test subjects' faces to be seen. Because every human has their personal thoughts there still might be factors affecting the emotions captured and read by the FER engines. Therefore it cannot be said with complete certainty that the emotions captured by the facial expression recognition systems are solely by the impact of the user interface.

Since AI systems are trained with different although vast data sets, there can always be biases that are passed forward to the recognition systems. The purpose of this study wasn't to evaluate these possible biases, but it aimed to get results that were able to be validated and are trustworthy.

The types of facial expression recognition systems and AI that were evaluated in this thesis have multiple different use cases, and therefore different systems will also fit best to for different situations. Although this experiment studied how well the two current systems worked, there can be better use cases for each of the systems, where

their performance would be better. It should also be noted that AI is a rapidly developing field, which means that the current systems by these developers might be undergoing improvements frequently. Executing this study after some time using the same FER systems can therefore give slightly alternating results.

5 Results

The analysis of the results of the CSV files and the Google Forms CSV is done with the help of 12 Python scripts. The scripts were developed with the assistance of artificial intelligence programming tools, specifically Claude Sonnet 4.6. The scripts for the analysis can be found at `https://github.com/sarakeskilohko/thesis_analysis`.

5.1 Evaluation of Facial Expression Recognition Systems

The experiment had 22 successful runs out of all 25. The reason for dismissing the three runs was in each case network issues leading to no API responses and therefore no valid FER data. For the 22 successful runs 45.5% were completed with AWS Rekognition and other 54.5% with Google Vision. Randomization for the FER model provider in the application was responsible for these results.

The FER models of both providers analyzed frames they received every 6 seconds. Total number of frames that were analyzed were 1450 per provider. Frames were skipped when a face wasn't detected in the frame or when API calls occurred too often for the UI to stay adaptive and not confusing. This lead to 31% of frames of all runs to be skipped for both of the APIs. Since participants took different amount

of time to finish the exercises, time isn't a variable in the results and findings. Time is rather measured by progressing to the next stage in the math exercises.

The stress score calculations show that AWS Rekognition based stress scores were always higher than Google Vision based stress scores. The stress scores, their means, medians, min and max values are presented in Table 5.1. The correlation between calculated stress by using AWS Rekognition and Google Vision is nearly zero, and the normalized stress values are presented in Figure 5.1. The correlation was calculated with Pearson correlation coefficient, where the value $r = 0$ represents no linear correlation.

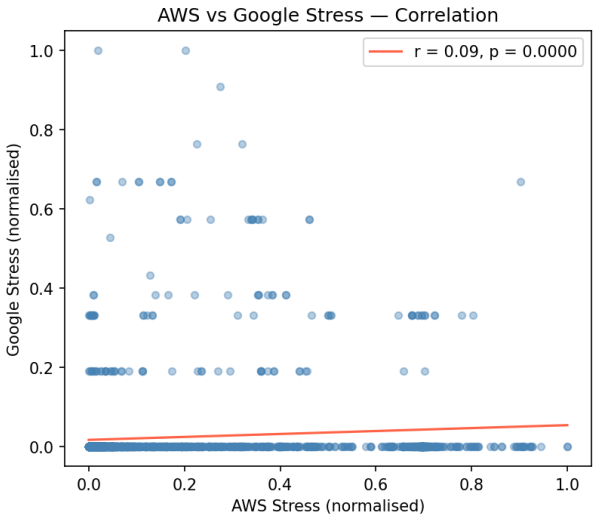


Figure 5.1: Correlation between AWS and Google calculated stress

The average stress value was highest at exercise stage 4, and at this stage the standard deviation was also the highest. The lowest stress scores were recorded at exercise stage 2, where the deviation also was at its lowest. This is visualized in Figure 5.2. Generally the stress scores didn't gradually increase as the participant progressed in the math stages. The mean and median values for stress scores throughout the math stages, as well as the standard deviation, are presented in Table 5.2. The distribution of mean values of stress scores by each provider can be

Table 5.1: Stress score data, scale (0-1)

	AWS Rekognition	**Google Vision**
Mean	0.0716	0.00609
Median	0.0130	0.0
Standard deviation	0.1060	0.0268
Min	0.0	0.0
Max	0.430	0.262

seen in Figure 5.3.

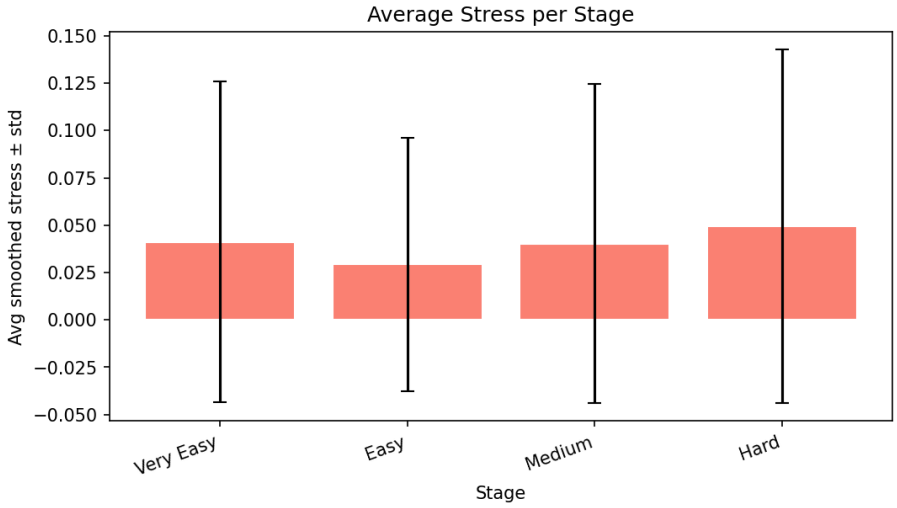


Figure 5.2: Average stress scores and standard deviation during different stages

The FER models' accuracies can be studied with self-reported information from the survey. The most detected emotions through all the stages were calm and joy. Highest correspondence with survey reported emotions and FER model detected emotions were for AWS Rekognition at around 70%. Detected emotions were mainly neutral

Table 5.2: Stress score statistics by stage and provider

Stage	Provider	Count	Mean	Median	Std	Min	Max
Very Easy	AWS	252	0.0683	0.00950	0.103	0.0	0.392
	Google	252	0.00996	0.0	0.0323	0.0	0.200
Easy	AWS	279	0.0632	0.0100	0.0981	0.0	0.388
	Google	278	0.00843	0.0	0.0327	0.0	0.262
Medium	AWS	269	0.0707	0.0140	0.108	0.0	0.389
	Google	269	0.00765	0.0	0.0320	0.0	0.262
Hard	AWS	282	0.0787	0.0140	0.112	0.0	0.430
	Google	281	0.00199	0.0	0.0121	0.0	0.100
Very Hard	AWS	368	0.0753	0.0175	0.107	0.0	0.397
	Google	367	0.00367	0.0	0.0204	0.0	0.175

or positive for both models, at 68.7% of frames contributing to this. Heatmaps of recognized emotions are presented in Figure 5.4, and Figure 5.6 shows the most detected emotions by wider categories.

5.2 Adaptive UI Behavior

Since the detected stress scores varied between AWS Rekognition and Google the amount of UI support level increments also varied. Figure 5.9 shows the differences between UI support level changes between providers per math exercise stage. No UI support level changes occurred in the 5th math exercise stage, and isn't therefore

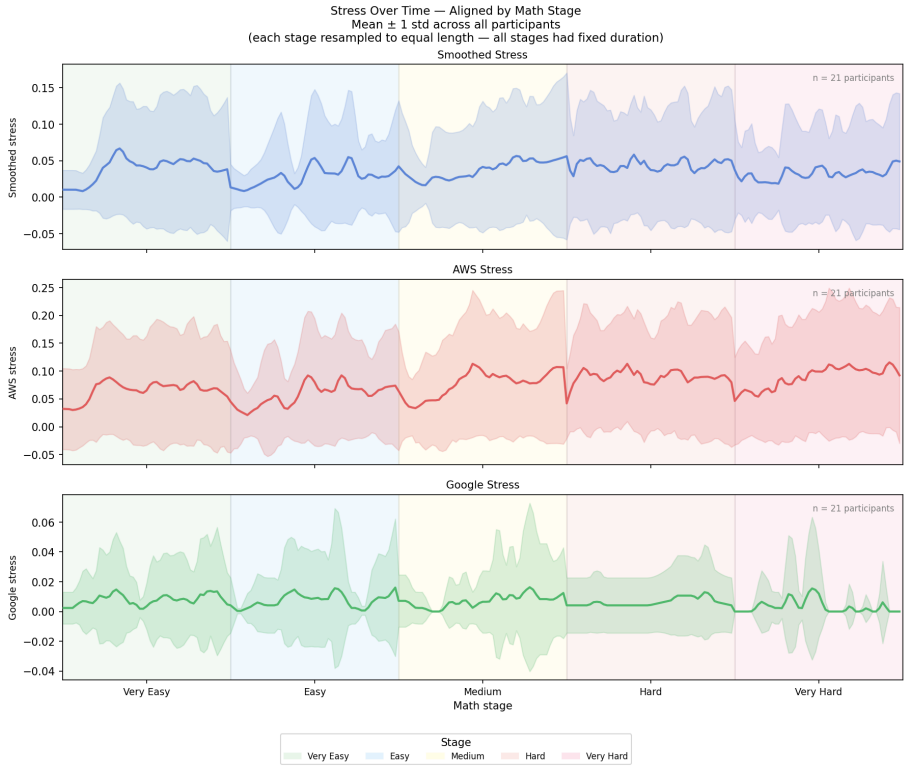
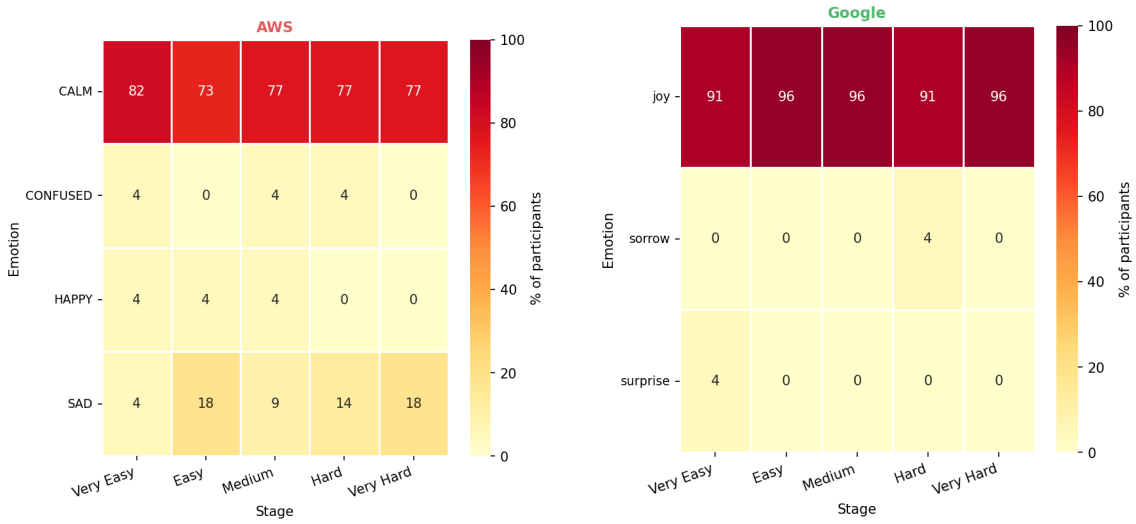


Figure 5.3: Mean detected stress per provider throughout runs



(a) AWS Rekognition (b) Google Vision

Figure 5.4: Most detected emotions per stage

seen on the figure. Overall UI support levels increased more than twice as much with AWS Rekognition than with Google Vision. Seven of the runs with Google Vision

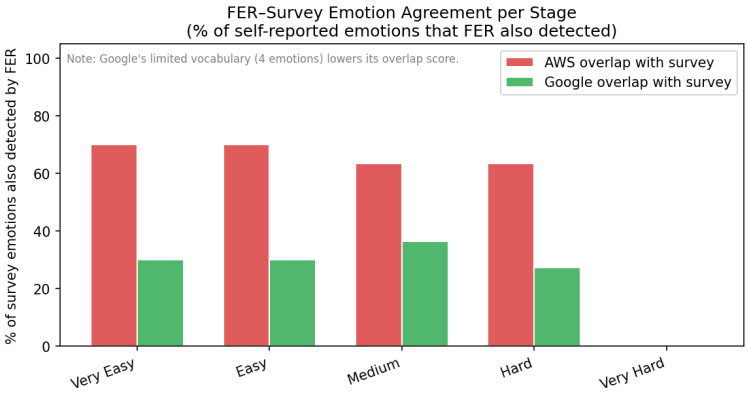


Figure 5.5: Detected emotions against survey reportings

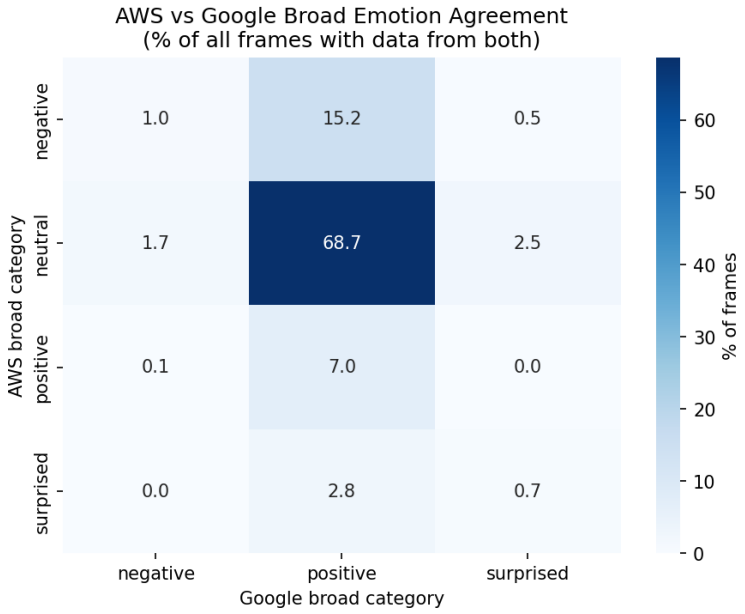


Figure 5.6: Most detected emotions for the same run

the participant's UI stayed at the lowest support level, and only one participant using Google Vision as their FER provider reached support level 12, the highest level. For AWS Rekognition nine participants reached the highest UI support level. Altogether 45% of runs were able to reach the highest possible support level, as can be seen on Figure 5.7. The numerical data on what was the highest UI support level that was reached is displayed in Table 5.4. Figure 5.8 shows that the number of level change trigger events was much higher with AWS Rekognition than the number of

triggered events with Google Vision. At most Google Vision was able to achieve half of the level change events that were triggered with AWS Rekognition. The specific amounts of triggered level change events per support level can be seen in Table 5.3.

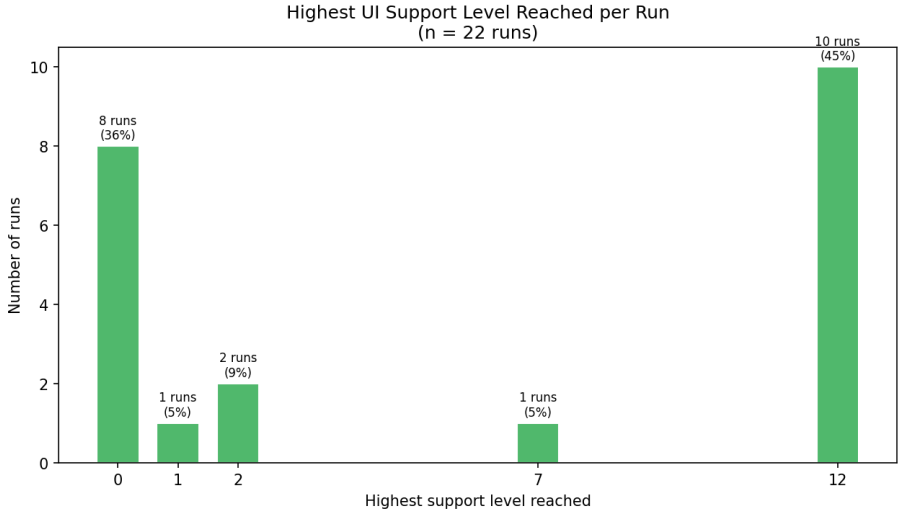


Figure 5.7: Highest support level reached during runs

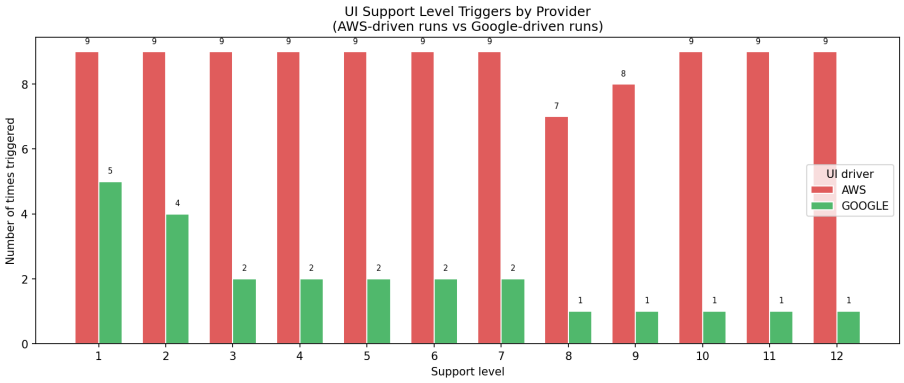


Figure 5.8: Amount of triggered UI support level increments per FER provider

Each support level was triggered at least once, and for example the first support level after the starting level 0 was triggered 63.6% of the runs. The highest support level was triggered in just 45.5% of the runs.

For 36.4% of the runs the only UI support level that was reached was the initial level 0, and the participants didn't receive any UI adaptation. For 45.5% of participants

Table 5.3: UI support level trigger counts

UI support level	Trigger count	Percentage
1	14	63.6%
2	13	59.1%
3	11	50.0%
4	11	50.0%
5	11	50.0%
6	11	50.0%
7	11	50.0%
8	8	36.4%
9	9	40.9%
10	10	45.5%
11	10	45.5%
12	10	45.5%

the UI support level was able to reach the highest possible option.

Google Vision triggered seemingly much less UI support level increments than AWS Rekognition did. Average stress detected by AWS Rekognition was also higher than that detected by Google Vision. For AWS Rekognition the average stress score was around 0.075, where as Google Vision's detected average stress score was under 0.01. Average support level that was present during runs with AWS Rekognition was support level nine, and for Google Vision the average was support level two. The

Table 5.4: Highest support level reached per run

Max level reached	Runs	Percentage
0	8	36.4%
1	1	4.5%
2	2	9.1%
7	1	4.5%
12	10	45.5%

contrast between the two providers, the stress scores and the average UI support levels reached is made clear by Figure 5.10, which shows AWS Rekognition's detected smoothed stress scores at nearly 0.08, whereas Google Vision doesn't reach 0.01.

The highest average stress scores were detected when the participant had support level four visible, and the lowest average stress scores were recorded at support level two. The standard deviation in detected stress scores was at support level nine, and the least deviation was detected at the highest support level, level twelve.

5.3 Task Performance

Overall performance of all participant throughout different math exercise stages can be seen in Figure 5.13. From the plot it can be seen that there is a considerable amount of variation from stage 1 (very easy) to stage 3 (medium), and after this variation decreases. This means that in the earlier stages the participants performed very differently regarding correct answers, whereas in the last two stages nearly all participants performed nearly equally well. The biggest amount of variation can

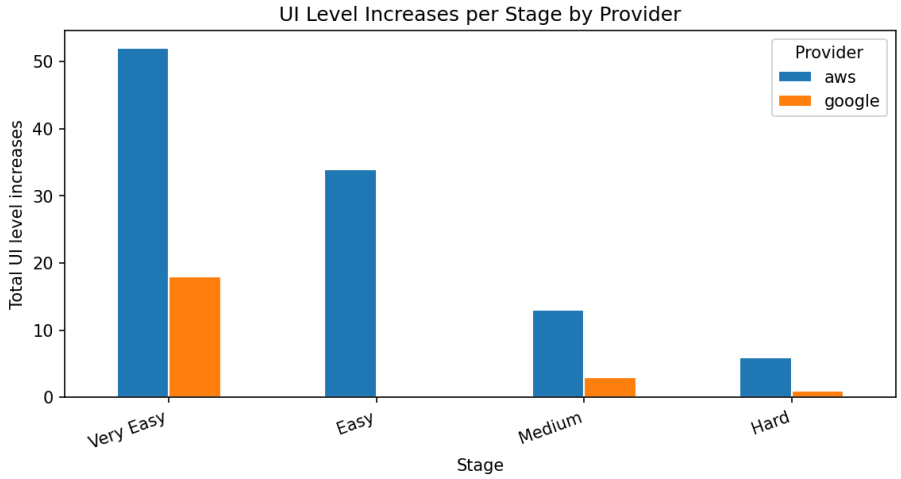


Figure 5.9: UI support level increments per FER provider

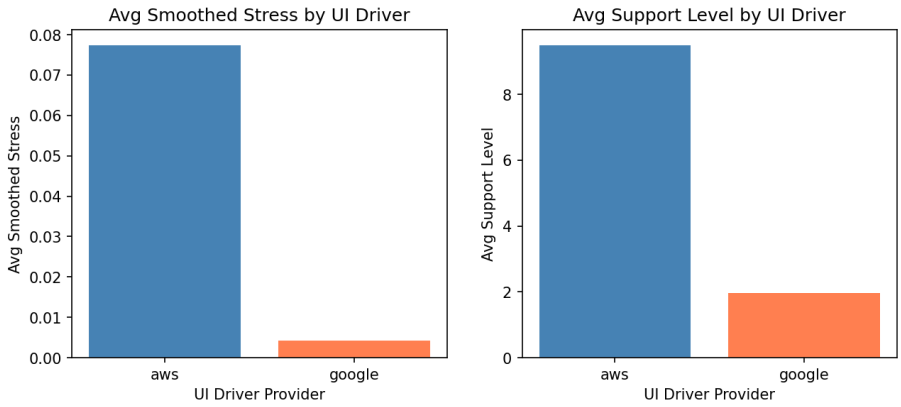


Figure 5.10: Average stress scores per stage and FER provider

be seen in the medium stage of exercises, where some got zero correct answers and others nearly 100% accuracy.

Figure 5.14 shows that the mean accuracy is higher when the UI was adapted than it was with no adaptation. The standard deviation is also increased when the adaptation occurred, meaning that more participants scored differently during the math stages.

The connections between self-rated math skills and correct answers through different stages can be seen in Figure 5.15. The detected average stress levels through different

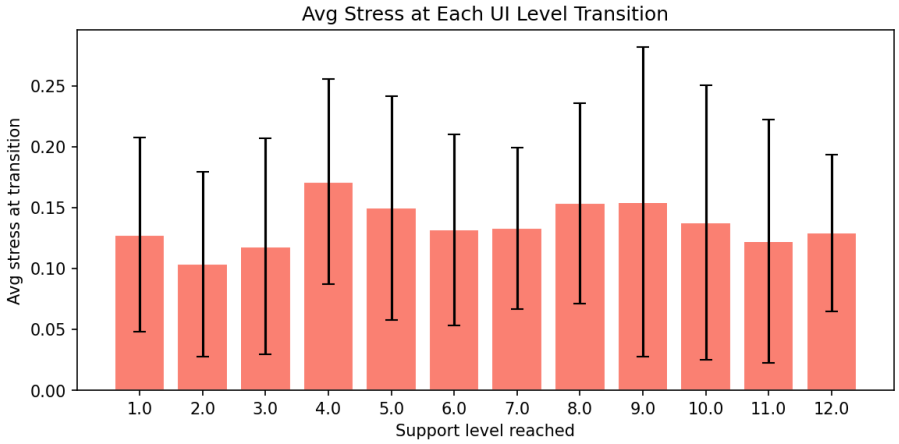


Figure 5.11: Stress levels per UI support level

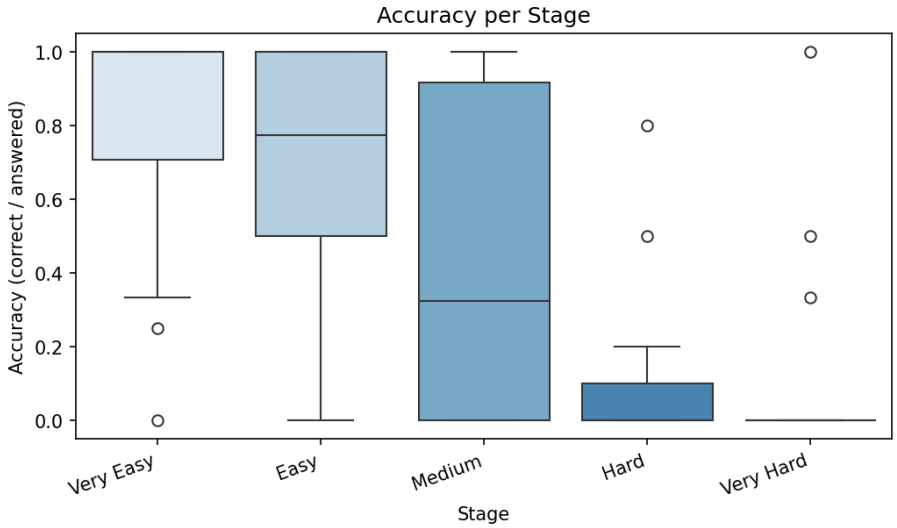


Figure 5.12: Accuracy throughout stages

math stages categorized by the math abilities that the participants reported having can be analyzed from Figure 5.16. Those who rated their skills to be at the highest level averaged a 0.0 stress score, whereas the highest stress scores were reported by those who rated their math abilities to be at level two out of five. However in Figure 5.17 it can be seen that those who rated their math abilities to be at the highest level also reported feeling the most stressed through all the math stages, whereas those who rated their skills to be at the lowest level reported having nearly same

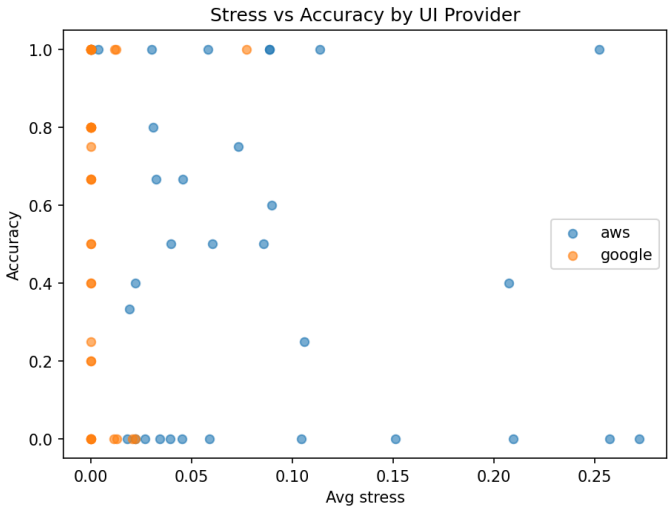


Figure 5.13: Stress scores and accuracy on performance

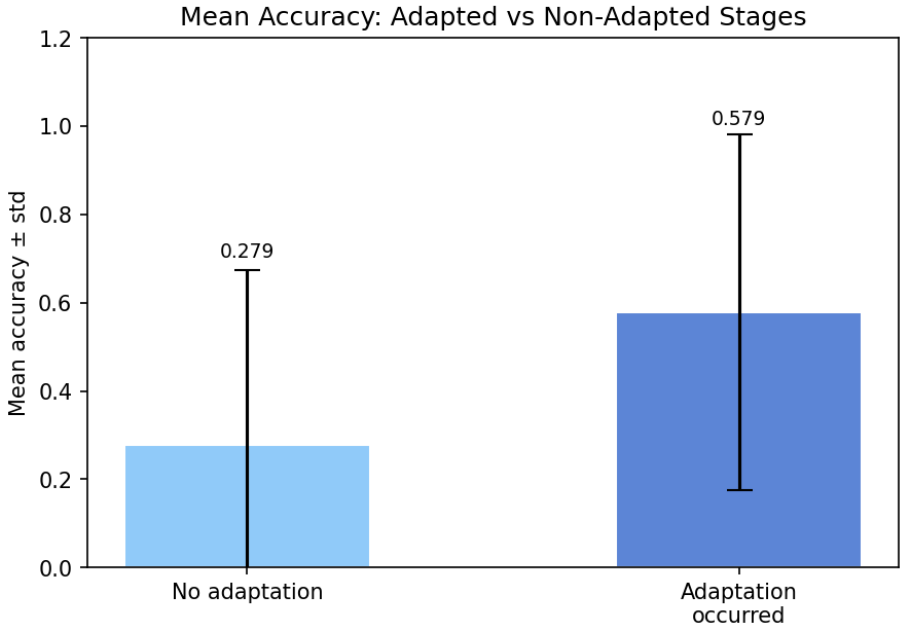


Figure 5.14: Accuracy scores regarding UI adaptation

level of stress during the experiment.

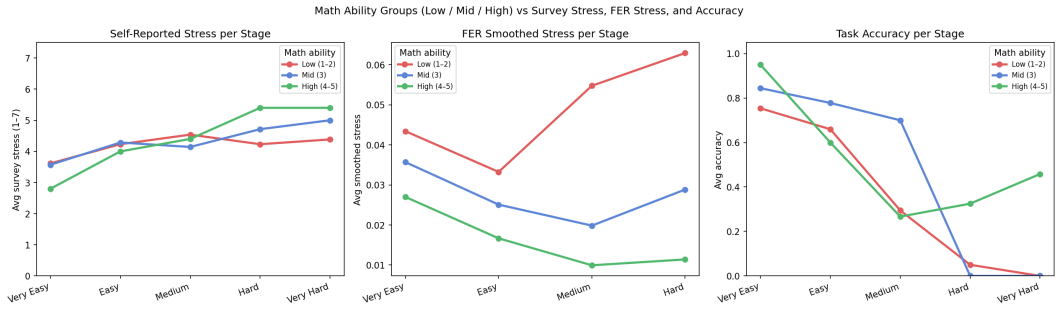


Figure 5.15: Self-rated math skills, stress levels and performance

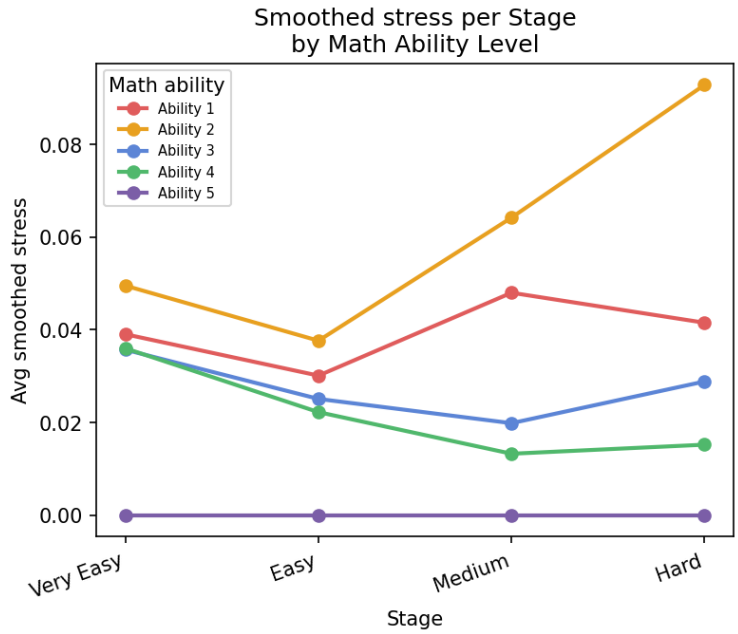


Figure 5.16: Self-rated skills and detected stress per stage

5.4 Survey Reports

The self-rated math abilities varied within the participants, and the distribution can be seen in Figure 5.18.

40% of participants reported feeling that the difficulty of the exercises increased gradually, and 64% reported feeling very time pressured during the experiment.

The self reported survey findings state that 53% of participants found stage 5 to be

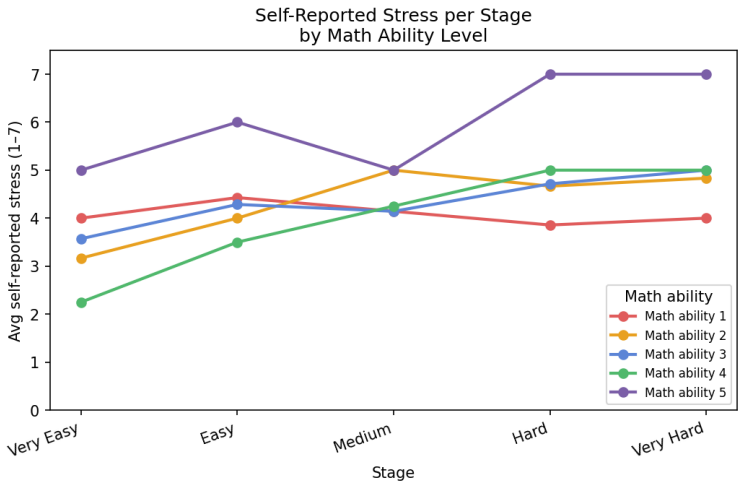


Figure 5.17: Self-rated skills and self-rated stress per stage

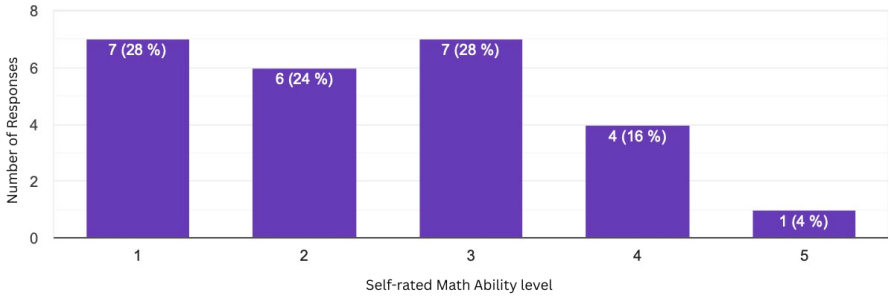


Figure 5.18: Self-rated math abilities

the most difficult one, and 36% found stage 4 to be the most difficult.

60% of participants reported noticing no changes in the UI during the exercises, but only 31% of participants actually had no changes.

Out of those who noticed change 50% reported feeling more supported when the UI changed, but 50% also stated not feeling more comfortable completing the exercises after the changes. 37.5% of participants who noticed change reported the change decreasing their frustration, and 37.5% reported feeling somewhat distracted by the UI changes.

During the experiment runs most participants reported the emotions they felt the

most to be focused. Many also reported feeling calm. The amount of answers for frustration are the highest at the third math stage. All participants reported being either somewhat aware of their emotions or very aware of them. 44% of participants reported the most negatively affecting factors in the experiment being difficult calculations, and 40% reported time limit being the most negatively affecting factor. More detailed reportings are presented in a heatmap in Figure 5.19.

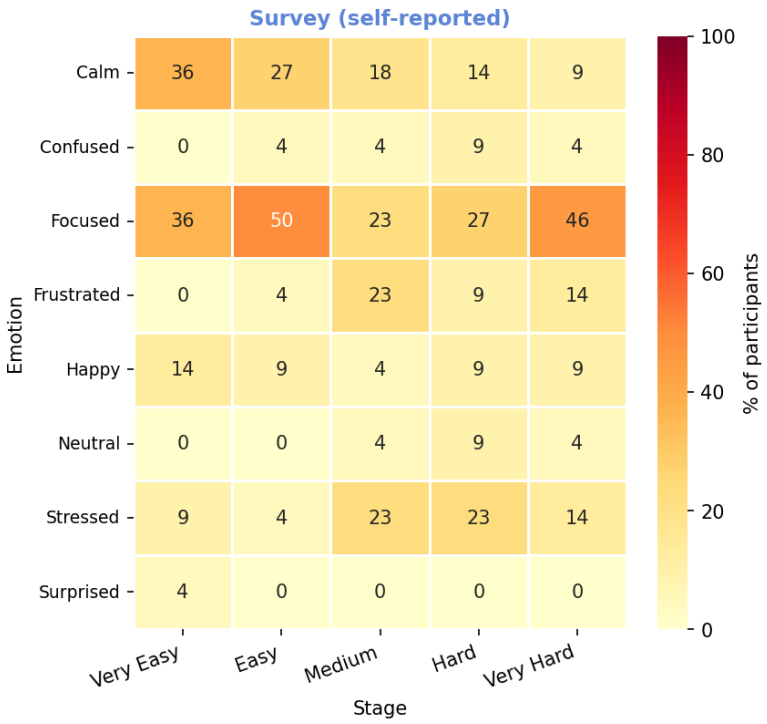


Figure 5.19: Most survey-reported emotions per stage

6 Discussion

6.1 Findings

6.1.1 FER in Creating a User Experience

The findings of the experiment suggest that adaptive user interfaces can improve user experience, but that it doesn't improve the experience for all participants. The two providers of FER models used in the experiment have immense differences in their accuracy and ability to recognize specific emotions. Neither of the models is on their own enough to create an adaptive UI that feels natural and responses to changes in facial expressions without latency or notable delays. There is still promise behind the findings to keep improving emotion-aware systems to respond to specific signals in real-time.

From Figures 5.3 and 5.11 it can be seen that even with UI adaptation, the stress scores didn't generally decrease once UI adaptation took place. On UI support levels 9, 10 and 11 the deviation between detected stress scores was the highest, so at these higher support levels, according to the FER models, some people felt very stressed while others were at ease. From 5.3 it can be derived that AWS Rekognition's detected stress scores increasing throughout the experiment even when UI adaptation occurred, whereas Google Vision's detected stress scores stayed more

consistently around 0, meaning that the participants weren't feeling more or less stressed. As can be seen from the Table 5.2 the median stress score for all the math stages in Google Vision was 0.0.

The reason for stress scores increasing for some participants even when UI adaptation happened could be explained by the increasing difficulty of the exercises the participants were to complete. For some participants the exercises might have been difficult enough so that no comforting changes in the UI could have decreased the stress. Participants also reached the higher UI support levels at different points of their runs, and for some the UI barely adapted due to low detected stress.

The stress smoothing choice made to improve the UX by limiting erratic changes in the UI due to high stress scores in few frames proved to be the correct choice, as can be seen in Figure 5.3. As the figure shows there are spikes of high stress scores seen on the graphs that AWS Rekognition and Google Vision detected. The smoothing ensured that the UI to reacted more gradually instead of jumping to a higher UI support level too fast, causing a distraction to the participants. Since only 40% of participants reported noticing any change in the UI the stress smoothing can be considered to have helped in the gradual increase of the level, therefore improving the UX altogether.

The system as a whole was a HCI application, where emotion-awareness was pursued. The UI adaptation was achieved when the FER models detected stress, although the stress detection itself varied widely between the two providers. Around 10% of participants reported the UI to be the factor that affected their emotions positively the most during the study, so it can be said that the adaptation improved the UX of at least 10% of the participants. The FER models however didn't accurately recognize all emotions that were expressed by the participants, which ultimately tells that for a truly emotion aware system more modalities should be used to correctly

recognize the emotions of the user. However there are great differences even between the two providers used in the experiment of this thesis, so choosing the most accurate FER model can also help with improving emotion-awareness and therefore the UX.

6.1.2 Evaluation of FER model accuracy

From Figure 5.5 it's visible that AWS Rekognition more closely aligns with what the participants reported feeling throughout the experiment runs, since around 70% of participants reported the same emotions AWS Rekognition detected, and only a little over 30% of participants' reported emotions corresponded with Google Vision. This means that AWS Rekognition can more accurately detect emotions, although it's success rate still isn't 100%.

AWS Rekognition and Google Vision both detected mostly neutral emotions, although Google Vision doesn't include a true neutral emotion, but rather the most frequently detected facial expression was "joy". For both of the FER models these neutral or happy emotions were detected the most, so there can be some similarities in the ways the models detect emotions. This can be seen in Figure 5.4. Still the difference between average stress scores tells that the emotions that are used in calculating the stress score are detected in different ways. These emotions are negative ones, for AWS Rekognition angry, sad, confused and surprised, and for Google Vision anger, sorrow and surprise. The emotion output with confidence scores and likelihoods can also affect the score calculation. Especially since Google Vision's likelihoods are given at a five-point scale the normalization for the detected emotion can be less accurate than AWS Rekognition's scale of confidence $(0 - 100)$.

Even though there are similar emotions detected at the runs in general and the categories of detected emotions are somewhat linked, the calculated correlation between the two providers during the same runs is actually only 0.09, as can be seen from

Figure 5.1. The figure depicts that Google Vision's normalized stress values stayed the most at a 0.0 score, meaning that no stress was detected. This has to do with the limited emotions that the FER model can recognize, as well as the limited scale on which the predictions are given. Table 5.2 shows the mean and median values for stress scores through the different math stages, as well as standard deviation. It can also be seen that AWS Rekognition always reached higher stress scores than Google Vision did. If both of the FER models produced the output more similarly or were able to categorize the same emotions, the correlation might be better. Choosing FER models that are more similar with each other could've improved the correlation as well. Different type of normalization could have also changed some values, but overall the bigger differences come from the predictions based on continuous confidence scores and the five-point likelihood scale and their differences. The fact that Google Vision produced stress scores of around 0.0 much more than AWS Rekognition is also behind the fact that participants who got Google as their FER model provider barely received UI adaptation, since stress scores always stayed low.

As stated in previous chapters many databases and datasets used in training and testing FER models can lead to biased results, depending on how diverse and robust the data is. Therefore in addition to the confidence scores and the five-point scale the models can be differently biased. They could have been trained and tested on different databases, and since the technology behind the APIs isn't available to the public, it isn't possible to know what kind of biases these databases might have. This bias might lead to different results for different participants, but with the amount of data that was gathered from this experiment it's not possible to rate how the bias might be showing in the results.

As for the FER models used in the experiment it's clear that AWS Rekognition provided more accurate representation of emotions. The choice of the provider

makes a difference in an experiment setting such as this, and when designing an application utilizing facial expression recognition it is crucial to choose the FER provider carefully to get the best results. For an emotion-aware, adaptive application AWS Rekognition could on its own be a working addition, but only if it's ensured that the users face forward and keep their faces visible to the camera, so that all faces and facial expressions can be detected. On the other hand Google Vision would not be sufficient for an adaptive application, and it would need to be customized to include more emotion categories, better normalized to handle the output data and be used in addition to other modalities for detecting stress levels and emotions correctly.

6.1.3 Constraints of the Experiment and FER Models

Constraints within the experiment for using FER models came from example from skipped frames that were not used in the calculation of stress scores. Frames were skipped when no face could be detected by either of the FER models, which in the end led to almost a third of all frames being skipped in the FER analysis and stress calculation. This speaks for the fact that the user's face must be clearly visible for the camera feed to capture accurate emotions in real-time. When the user expresses emotions but their face is for example turned away enough so that the model isn't able to detect a face, the expressions cannot be analyzed either. This means that for an accurate, emotion-aware application utilizing just facial expression recognition the user has to concentrate on keeping their whole face visible to the camera. Having to pay attention to these details can in general have negative effects on the UX. In the application of the experiment when no face was detected the stress score for that frame was kept at the same value that was detected previously, but in real life the stress scores could have changed if a face could be detected at all times. In the experiment it was seen that the skipped frames occurred mostly when participants

looked down on the table where the laptop was located or moved to either side.

One of the biggest limitations with the tested FER models are with Google Vision's ability to recognize emotions. Since the model can only recognize four emotions it can produce very limited results for detected facial expressions. This was clearly seen in the experiment too, since the emotion correspondence with what the participants reported feeling matched much more closely with AWS Rekognition than with Google Vision. For a robust application a sufficient FER model is needed, and for this purpose Google Vision wasn't sufficient enough.

The way normalization was done for Google Vision can also affect the scores. The numerical values that were assigned to each of the five stages were $0, 0.25, 0.5, 0.75$ and 1.0, meaning that the gaps between each likelihood are equal. That means that for example `LIKELY` and `VERY_LIKELY` are as near each other as `UNLIKELY` and `POSSIBLE`. The stress scores calculated with AWS Rekognition and Google Vision in the experiment might have correlated better if the normalization scale was designed differently.

The evaluation of the FER models is studied by math stage and by UI support level, not frame by frame. Therefore it's possible that in a certain math exercise stage of at a certain UI support level the participant has experienced multiple different emotions and expressed them, but the FER models only receive a limited number of frames, and the results are analyzed mostly by the stage of by the support level, since the frame count is too high to analyze them frame by frame. The participants also wouldn't have been able to report their actual emotions per each frames, i.e. every six seconds, so there is no way to know for sure if the detected emotions could correspond frame by frame with what the participants felt. Even though the application aimed to be real-time there was always some latency, so the UI adaptation couldn't occur at the exact moment the stress was felt or detected. The

latency is an issue with all real-time FER models that are used through APIs.

6.1.4 Impact of an Adaptive User Interface

From the self-reported information from the survey it can be seen that for 50% of the participants the adaptation started at the easiest stage, so high enough stress scores were achieved already in the first stage. Two participant received the first adaptation at stage three, and one at stage four. It can be derived that those who felt the most stressed started expressing their stress already in the first stage, and those less stressed expressed in later on or not at all. No one experienced adaptation in the final stage, either the highest support level was already reached by this point or the participants didn't express stress enough for the support level to adapt anymore.

Only 40% of participants reported noticing changes in the UI. Truly for 69% of the participants the UI adapted, so multiple participants didn't notice the changes even when they occurred. The changes were designed to be slight as not to distract the user, and therefore it's not a negative result that the users didn't notice these changes. The changes in the UI could've still affected the stress levels of the participants subconsciously.

Besides adaptation in the UI the more stressed the participants felt the worse their UX could have been. Based on the survey reportings the level of mathematical abilities was somewhat linked to how stressed the participants felt, as can be seen on Figures 5.16 and 5.17. Based on the self-rated stress levels it seems that those with the highest rated abilities in mathematics also rated their stress to be the highest, while other rated levels of abilities follow a similar trend, the lowest rated abilities being a little higher rated on stress levels. However the detected stress scores implicate that participants who rated their abilities lower experienced more stress, and those who rated their abilities to be higher scored significantly lower

stress scores as well. This suggests that the abilities to complete the exercises was connected with how stressed the participant felt, and therefore could have affected the UX.

In Figure 5.12 it can be seen that the most accurate answers are given at the easiest stage, and the accuracy decreases as the stages progress. There is a lot of deviation in the answers at the second and third stages, where some participants have answered all exercises correctly, and some have answered none.

Altogether the accuracy decreased when participants progressed in the math stages. At the same time UI support levels progressed to higher ones. However Figure 5.14 suggests that after UI adaptation the accuracy still increased. This can be because in most runs the first adaptation occurred already in the first math stage, meaning that the participants' answers were only done on adapted UI. Therefore it cannot be said that adaptation helps with performance on the exercises, but since the exercises also got more difficult through the stages that also affects this finding. The mean accuracy for a stage where adaptation happened was 0.78, and the mean accuracy before the adaptation was 0.89. The mean accuracy for after the adaptation was 0.5. The accuracy decreases but the difficulty might be more to blame here, rather than the UI adaptation.

As stated earlier, around 10% of participants listed the UI as the biggest factor for positive feelings. Therefore it can be argued that at least for some participants the changes in the UI made a difference, and they had an improved UX. However around 15% of participants reported the UI changes being somewhat distracting, so the UX was worse for some participants because of the adaptation. At the same time 15% of participants reported feeling less frustrated when the UI adaptation occurred. 20% of participants reported feeling somewhat more comfortable after the UI adaptation, which can also be considered towards a more positive UX.

It can be interpreted that the UI adaptation affected the UX for better and for worse, depending on the participant. To say clearly which one of these is the more popular direction there would be need to conduct more studies on a larger sample group. The self-rated stress scores somewhat correspond with the scores calculated based on the FER models, as can be seen in Figure 5.15. The participants overall reported feeling more stressed the further the stages progressed, and that was the consensus with the FER models as well, specifically for the smoothed stress scores. Therefore it can be said that while the FER models were unable to correspond with the actual emotions of the participants accurately they still follow the same trend in the increasing stress scores.

The usability standards introduced in chapter two state that usability is effectiveness, satisfaction and efficiency. Satisfaction with the application and the user interface can be analyzed through the survey reportings. 48% of the participants stated that the factor for their positive emotions during the experiment was succeeding in the exercises and understanding them. This can be interpreted so that this percentage of participants felt at least some what satisfied with the application. Even when they reported feeling more stressed throughout the levels they found something that increased their positive emotions. 10% of participants also reported feeling supported and comfortable when completing the exercises, which also indicates satisfaction. As for effectiveness the accuracy of completing exercises decreased in general when proceeding to the harder levels, but this was also the way the system was designed, so the accuracy not improving while the UI adapted wasn't a complete impossibility. Measuring efficiency wasn't ideal for this experiment, since all participants either used all the time they had available for the exercise, or gave up and submitted no answers before the timer ran out. Therefore efficiency is difficult to evaluate. The efficiency of the UI adaptation itself can always be improved, since latency with the external API makes the whole adaptation process less efficient,

since the adaptation always occurs with a delay.

All in all the FER models' output and adaptation accordingly can be a supportive addition to user interfaces and the way they adapt to the users' preferences, but as of now there is still the need to study the effects more closely. A FER model itself has to be carefully selected if accurate emotions have to be detected, and mitigating the issue with delays and latency can prove to be difficult. A FER model in itself might not be enough to get an accurate reading of the user's state of mind, and having to use and pose for a camera creates issues with spontaneous and natural situations and environments. Adaptive user interfaces themselves can offer support for certain users, but not everyone is fond of them, which is why there's still the need to examine how the adaptation can be done so that it suits each user's preferences.

6.2 Constraints and Limitations

The limitations in the experiment have to do with the sample size, since only 25 participants completed the experiment. Therefore the findings might not be generalizable to the public. Some of the participant shared a similar background, and the sample group could have benefited from being more diverse regarding background and age.

The black box problem can cause constraints on an experiment such as this one. Since there is no information about what kind of datasets were used to train either of the FER models used in the experiment, it is unknown if the FER model predictions contained bias, or if they would've respond better to posed facial expressions, since most databases contain posed training data. Since at best the FER models' detected facial expressions corresponded with reported emotions at around 80% match it is clear that not all emotions were detected correctly by either of the providers. That being said both the environment and the FER models themselves might need

improvement for more accurate results.

The stress caused by the exercises in the experiment might not reflect real-life applications where FER models and adaptation could be utilized to improve UX. The experiment rather created a high-stress situation where the stress was designed to increase quickly, and the participants were aware of the camera. Knowing that frames are taken from the video feed the participants could've been more aware of expressing their emotions, which might not correspond with a spontaneous, natural setting. The difficulty of the exercises could also have been too difficult for some of the participants, therefore causing them to give up and no longer expressing stress in the experiment.

The experiment lasted altogether around 10-25 minutes depending on the specific participant. Since this is generally a short amount of time the effects of an adaptive UI might have been better seen at a longer experiment, where the stress scores could have fluctuated more gradually too.

With the FER models the most significant constrain came from Google Vision and its ability to recognize different emotions. Since it can only detect four emotions the category "joy" was clearly the emotion that was detected the most during the experiment runs. This is because Google Vision doesn't include a "neutral" emotion that can be detected, so the closest one would be joyous. Besides the lack of recognizing different facial expressions the output given on a five point scale limits the data. The data has to be normalized to be analyzed or studied further, and the normalization has to be done by splitting the five point scale in some way, and this leads the numerical values to be set with specific factors representing the likelihoods.

6.3 Future Work

For future reference there is a need to conduct more studies. A sample group should be larger than the 25 participants that was used in this experiment, since deviation was high between participants. Only a smaller percentage of participants noticed changes in the UI, and future studies would benefit from more self-reported emotions and feelings to further evaluate the FER models and the way adaptation affected the users. The sample group should be chosen to be diverse, and the tasks the participants should have to complete other kinds of tasks besides math exercises. To truly see how the adaptation of the UI affects the participants the experiment runs could be longer to see the affects in a longer run. A similar application could also be tested in a more spontaneous environment, where the camera isn't directly in front of the user, to see how the FER models perform in a situation that more likely mimics real life.

Another factor that might affect the participants' emotions and therefore their attitude toward the experiment are the feelings and events previous to the experiment. The experiment in this thesis did no research into what the participants felt like before they started the experiment, and the FER models also started to analyze frames only after they were made aware of the exercises. Underlying emotions can affect what the participants express, and therefore future studies could take into account what the participants feel like before the experiment run, and how the UX is affected by that.

In general based on the experiment it can be said that FER models that produce the output with confidence scores are more accurate and easier to utilize than those that give the results on a specific point scale, with no continuous values between the points. It could also be beneficial to try to train a custom model for recognizing facial expressions. The custom model could be run locally, and it could be studied

if latency and delay could be minimized this way. With a custom trained FER model there is no black box problem, and any biases can be known to the study conductors. In addition to this a more frequent loop of frame analysis could be tried to see, how fast the UI is able to adapt to the emotions expressed by the user. Since the experiment application prohibited the API calls to be at most every six seconds, the response to facial expressions wasn't as real-time as it could have been with more frequent API call loops.

For a more robust way to acquire information about the users' emotions more modalities should be used. For example hear-rate monitors might give more insight into how stressed a user is feelings.

7 Conclusion

As the utilization of AI in UI design and adaptation is increasing the need to design application to be emotion-aware is a possibility that can lead to improved UX. One way to incorporate emotion-awareness in UX improvement and UI adaptation is through utilizing facial expression recognition. This thesis included designing and implementing a web application with an adaptive UI, that responded to signals gotten from two different FER models. The experiment also evaluated the accuracy of these models.

The application for the experiment was an adaptive web application that utilized AWS Rekognition and Google Vision APIs simultaneously. The application received frames from the device camera feed, and the APIs analyzed the images, producing an output with categorized emotions. Based on the outputs a smoothed stress score was calculated from the negative emotions that were detected, and once the stress score reached a certain threshold, the application adapted, aiming to prompt more positive feelings. Altogether the application had 12 different UI support levels that aimed to improve the UX while the user progressed in the application. In the experiment the participants needed to complete math exercises that gradually got more difficult while the FER models analyzed their facial expressions. 25 user took part in the experiment, and the FER outputs and participants' process in the experiment were logged in CSV files for further analysis.

The first research question aimed to discover how facial expression technology can be used in creating a positive user experience. Based on the findings of the experiment FER models can be used in creating adaptive, emotion-aware applications. The FER models have to be sufficiently accurate in order for the UI to reflect the true feelings of the users, and real-time adaptivity is key to keep the UI responding to rising stress levels, therefore providing comfort and improving the UX.

On the second research question investigating how accurately the currently available FER models recognize facial expressions, the findings show significant differences between models from different providers. The experiment evaluated the accuracy of AWS Rekognition and Google Vision, and the findings show that AWS Rekognition corresponded with the emotions reported by the participants up to accuracy of 70%, whereas Google Vision's output corresponded with just around 30% accuracy with the self-reportings. There was barely any correlation between the facial expressions most recognized by the different FER models during the same runs, although both models recognized neutral or positive emotions the most. This speaks for the fact that the choice of the FER model provider affects the results of an experiment greatly.

Regarding constraints of the FER technologies it was found that the amount of emotions that can be recognized as well as the way the output if produced affects the results. From the calculated stress scores it could be seen that AWS Rekognition which gave the output on eight different emotions and the confidence score (0-100) on specific emotions, the stress scores corresponded the self-reported stress levels much more closely, than the scores calculated based on Google Vision's output. This is a result of Google Vision only being able to detect four emotions, and the likelihood of the emotions is given on a five-point-scale, that had to be normalized to be able to be comparable with the AWS Rekognition results. Both of the models weren't

able to detect faces and therefore neither the facial expressions if the user's head wasn't facing the camera, which lead to frames being skipped from the evaluation.

The final research question examined how an adaptive interface can affect the UX. The findings state that around 20% of participants gained comfort from the adaptation of the UI. Around a third of participants noticed no adaptation in the UI, although the UI adapted to their stress levels. This speaks for the fact that the adaptation was able to occur without it being distracting. On average the stress scores increased while the experiment progressed, even with the UI adaptation providing comfort. This is explained by the fact that the content of the exercises also increased greatly in difficulty, prompting more stress. Still, 10% of participants listed the UI to be the factor for positive emotions during the experiment, which can count towards an improved UI. To see how UI adaptation affects stress scores in the long run more studies with larger sample groups are needed to be conducted.

This study provides a functional web based application where adaptation is based on the output of two FER models. This proves that an emotion-aware, adaptive application can be designed and implemented around cloud based APIs, even though the APIs have to be chosen carefully, and latency in the adaptation frequency has to be considered. It also speaks for the fact that existing APIs have significant differences in their accuracy, and the choice for the correct FER API for the purpose has to be considered carefully. Specifically the differences between AWS Rekognition and Google Vision outputs from the same data are evaluated in the study. These contributions increase the understanding of FER based adaptive applications and the accuracy and limitations of AWS Rekognition and Google Vision, working as a base for future work.

The findings of this thesis suggest that FER models can be utilized by incorporating them into adaptive applications, but they still cannot be trusted to recognize all

facial expressions or emotions. To create a more emotion-aware application there is a need for more modalities to be used with the FER models, and the FER models themselves should be improved to recognize emotions more accurately. Especially the shortcomings of Google Vision speak for the fact that FER technology could and should still be improved to achieve accurate results that can better be utilized in applications. Since most of the available databases are trained with datasets consisting of posed images the models still have a hard time recognizing spontaneous facial expressions, and constructing more robust and diverse databases is something that would benefit the field in the future. Transparency in how and with what databases FER APIs are trained would improve mitigating biases in the recognition systems, and standardizing how outputs are produced would ease integrating APIs from different providers into one application. While this thesis can be used as a foundation for future work in the field of utilizing FER models in adaptive user interfaces there is still a need for more studies around the topic.

References

[1] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields", *Electronics*, vol. 11, no. 1, Jan. 3, 2022, ISSN: 2079-9292. DOI: 10.3390/electronics11010141. Accessed: Mar. 10, 2026. [Online]. Available: https://www.mdpi.com/2079-9292/11/1/141.

[2] D. Véras, T. Prota, A. Bispo, R. Prudêncio, and C. Ferraz, "A literature review of recommender systems in the television domain", *Expert Systems with Applications*, vol. 42, no. 22, pp. 9046–9076, 2015, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.06.052. Accessed: Mar. 10, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417415004546.

[3] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, and J. Basilico, "Deep learning for recommender systems: A netflix case study", *AI Magazine*, vol. 42, Nov. 20, 2021. DOI: 10.1609/aimag.v42i3.18140. Accessed: Mar. 10, 2026. [Online]. Available: https://doi.org/10.1609/aimag.v42i3.18140.

[4] J. S. Persson, A. Bruun, M. K. Lárusdóttir, and P. A. Nielsen, "Agile software development and ux design: A case study of integration by mutual adjustment", *Information and Software Technology*, vol. 152, p. 107 059, 2022, ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2022.107059. Accessed:

Mar. 4, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584922001690.

[5] Å. Stige, E. D. Zamani, P. Mikalef, and Y. Zhu, "Artificial intelligence (ai) for user experience (ux) design: A systematic literature review and future research agenda", *Information Technology & People*, vol. 37, no. 6, pp. 2324–2352, Aug. 2023, ISSN: 0959-3845. DOI: 10.1108/ITP-07-2022-0519. eprint: https://www.emerald.com/itp/article-pdf/37/6/2324/9569627/itp-07-2022-0519.pdf. Accessed: Mar. 10, 2026. [Online]. Available: https://doi.org/10.1108/ITP-07-2022-0519.

[6] Z. Lv, F. Poiesi, Q. Dong, J. Lloret, and H. Song, "Deep learning for intelligent human–computer interaction", *Applied Sciences*, vol. 12, no. 22, Nov. 11, 2022, ISSN: 2076-3417. DOI: 10.3390/app122211457. Accessed: Mar. 10, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/12/22/11457.

[7] C. Stephanidis and G. Salvendy, "Human-computer interaction: Foundations and advances", *CRC Press*, Sep. 28, 2024. Accessed: Mar. 10, 2026. [Online]. Available: https://doi.org/10.1201/9781003584292.

[8] C. Gumbheer, K. Khedo, and A. Bungaleea, "Personalized and adaptive context-aware mobile learning: Review, challenges and future directions", *Educ Inf Technol*, vol. 47, Feb. 12, 2022. Accessed: Mar. 4, 2026. [Online]. Available: https://doi.org/10.1007/s10639-022-10942-8.

[9] M. Benaida, "Developing and extending usability heuristics evaluation for user interface design via ahp", *Soft Comput*, vol. 27, Jan. 10, 2023. Accessed: Mar. 10, 2026. [Online]. Available: https://doi.org/10.1007/s00500-022-07803-4.

[10] X. Lan, J. Xue, J. Qi, D. Jiang, K. Lu, and T.-S. Chua, "Expllm: Towards chain of thought for facial expression recognition", *IEEE Transactions on Multime-*

dia, vol. 27, pp. 3069–3081, Apr. 3, 2025. Accessed: Mar. 10, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10948346.

[11] F. Talaat, "Real-time facial emotion recognition system among children with autism based on deep learning and iot", *Neural Comput & Applic*, vol. 35, Mar. 7, 2023. Accessed: Mar. 10, 2026. [Online]. Available: https://doi.org/10.1007/s00521-023-08372-9.

[12] R. Kumar, G. Corvisieri, T. F. Fici, S. I. Hussain, D. Tegolo, and C. Valenti, "Transfer learning for facial expression recognition", *Information*, vol. 16, no. 4, Apr. 17, 2025, ISSN: 2078-2489. DOI: 10.3390/info16040320. Accessed: Mar. 10, 2026. [Online]. Available: https://www.mdpi.com/2078-2489/16/4/320.

[13] F. X. Gaya-Morey, J. M. Buades-Rubio, P. Palanque, R. Lacuesta, and C. Manresa-Yee, *Deep learning-based facial expression recognition for the elderly: A systematic review*, Apr. 4, 2025. arXiv: 2502.02618 [cs.CV]. Accessed: Mar. 10, 2026. [Online]. Available: https://arxiv.org/abs/2502.02618.

[14] Å. Stige, E. Zamani, P. Mikalef, and Y. Zhu, "Artificial intelligence (ai) for user experience (ux) design: A systematic literature review and future research agenda", *Information Technology & People*, vol. 37, no. 6, May 31, 2024. Accessed: Jan. 8, 2025. [Online]. Available: https://doi.org/10.1108/ITP-07-2022-0519.

[15] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects", *Journal of Industrial Information Integration*, vol. 23, p. 100 224, Sep. 2021, ISSN: 2452-414X. DOI: https://doi.org/10.1016/j.jii.2021.100224. Accessed: Jan. 13, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X21000248.

[16] J. Helm, A. Swiergosz, and H. e. a. Haeberle, "Machine learning and artificial intelligence: Definitions, applications, and future directions", *Curr Rev*

Musculoskelet Med, vol. 13, Jan. 25, 2020. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1007/s12178-020-09600-8.

[17] Y. Jiang, X. Li, and H. e. a. Luo, "Quo vadis artificial intelligence?", *Discov Artif Intell*, vol. 2,4, Feb. 28, 2022. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1007/s44163-022-00022-8.

[18] T. Song, L. Xuanyi, B. Wang, and L. Han, "Research on intelligent application design based on artificial intelligence and adaptive interface", *World Journal of Innovation and Modern Technology*, vol. 7, no. 2, Jul. 2, 2024. DOI: 10.53469/wjimt.2024.07(02).01. Accessed: Jan. 7, 2025.

[19] S. H. Shetty, S. Shetty, C. Singh, and A. Rao, "Supervised machine learning: Algorithms and applications", in *Fundamentals and Methods of Machine and Deep Learning*. John Wiley & Sons, Ltd, Jan. 29, 2022, ch. 1, pp. 1–16, ISBN: 9781119821908. DOI: https://doi.org/10.1002/9781119821908.ch1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119821908.ch1. Accessed: Mar. 19, 2026. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119821908.ch1.

[20] A. Hosna and J. e. a. Merry E.and Gyalmo, "Transfer learning: A friendly introduction", *J Big Data*, vol. 9, Sep. 19, 2022, ISSN: 102. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1186/s40537-022-00652-w.

[21] R. Dastres and M. Soori, "Artificial Neural Network Systems", *International Journal of Imaging and Robotics (IJIR)*, vol. 21, no. 2, pp. 13–25, Sep. 20, 2021. Accessed: Oct. 4, 2025. [Online]. Available: https://hal.science/hal-03349542.

[22] J. Naskath, G. Sivakamasundari, and A. Begum, "A study on different deep learning algorithms used in deep neural nets: Mlp som and dbn", *Wireless*

Personal Communications, vol. 128, Oct. 19, 2022. Accessed: Oct. 4, 2025. [Online]. Available: https://doi.org/10.1007/s11277-022-10079-4.

[23] J. Shao and Y. Qian, "Three convolutional neural network models for facial expression recognition in the wild", *Neurocomputing*, vol. 355, pp. 82–92, Aug. 25, 2019, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.05.005. Accessed: Oct. 5, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219306137.

[24] S. Sureddy and J. Jacob, "Multi-features based multi-layer perceptron for facial expression recognition system", *Second International Conference on Image Processing and Capsule Networks*, vol. Lecture Notes in Networks and Systems, vol 300, Sep. 10, 2021. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1007/978-3-030-84760-9_19.

[25] B. Yang, L. Wei, and Z. Pu, "Measuring and improving user experience through artificial intelligence-aided design", *Frontiers in Psychology*, vol. 11, Nov. 19, 2020, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2020.595374. Accessed: Jan. 7, 2025. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.595374.

[26] J. Š. Novák, J. Masner, P. Benda, P. Šimek, and V. Merunka, "Eye tracking, usability, and user experience: A systematic review", *International Journal of Human–Computer Interaction*, vol. 40, no. 17, May 31, 2024. DOI: 10.1080/10447318.2023.2221600. Accessed: Jan. 8, 2025. [Online]. Available: https://doi.org/10.1080/10447318.2023.2221600.

[27] J. Grigera, J. P. Espada, and G. Rossi, "Ai in user interface design and evaluation", *IT Professional*, vol. 25, no. 2, pp. 20–22, May 12, 2023. DOI: 10.1109/MITP.2023.3267139. Accessed: Mar. 7, 2025.

[28] Ł. Bielarczyk, "Meanings and interpretations of efficiency in iso standards", *SSRN*, Jul. 22, 2022. DOI: http://dx.doi.org/10.2139/ssrn.4169751. Accessed: Mar. 19, 2026. [Online]. Available: https://ssrn.com/abstract=4169751.

[29] E. Heinold, P. H. Rosen, and S. Wischniewski, "Usability questionnaire for robotic systems based on the iso 9241–110", *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2231–2238, Jan. 6, 2025. DOI: 10.1109/LRA.2025.3526557. Accessed: Mar. 19, 2026.

[30] K. Rodden, H. Hutchinson, and X. Fu, "Measuring the user experience on a large scale: User-centered metrics for web applications", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, Atlanta, Georgia, USA: Association for Computing Machinery, Apr. 10, 2010, pp. 2395–2398, ISBN: 9781605589299. DOI: 10.1145/1753326.1753687. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1145/1753326.1753687.

[31] J. Robinson, "Likert scale", in *Encyclopedia of Quality of Life and Well-Being Research*, F. Maggino, Ed. Cham: Springer International Publishing, Feb. 11, 2024, pp. 3917–3918, ISBN: 978-3-031-17299-1. DOI: 10.1007/978-3-031-17299-1_1654. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1007/978-3-031-17299-1_1654.

[32] S. A. C. Perrig, L. F. Aeschbach, N. Scharowski, N. von Felten, K. Opwis, and F. Brühlmann, "Measurement practices in user experience (ux) research: A systematic quantitative literature review", *Frontiers in Computer Science*, vol. Volume 6 - 2024, Mar. 4, 2024, ISSN: 2624-9898. DOI: 10.3389/fcomp.2024.1368860. Accessed: Mar. 19, 2026. [Online]. Available: https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2024.1368860.

[33] M. Schrepp et al., "On the importance of ux quality aspects for different product categories", *International Journal of Interactive Mul- timedia and Artificial Intelligence*, vol. 8, Mar. 13, 2023, ISSN: 232-46. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.25968/opus-3394.

[34] X. Xie, Y. Wang, Y. Cui, S. Yu, D. Chen, and J. Chu, "Evaluation of cognitive load and user experience in alternative interaction modes under different noise degrees", *Advanced Engineering Informatics*, vol. 65, p. 103 328, May 2025, ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2025.103328. Accessed: Mar. 19, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034625002216.

[35] S. T. Völkel, C. Schneegass, M. Eiband, and D. Buschek, "What is "intelligent" in intelligent user interfaces? a meta-analysis of 25 years of iui", in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, Cagliari, Italy: Association for Computing Machinery, Mar. 17, 2020, ISBN: 9781450371186. DOI: 10.1145/3377325.3377500. Accessed: Jan. 7, 2025. [Online]. Available: https://doi.org/10.1145/3377325.3377500.

[36] M. L. Seo-young Lee and G. Hoffman, "When and how to use ai in the design process? implications for human-ai design collaboration", *International Journal of Human–Computer Interaction*, vol. 41, no. 2, pp. 1569–1584, May 22, 2024. DOI: 10.1080/10447318.2024.2353451. Accessed: Mar. 7, 2025. [Online]. Available: https://doi.org/10.1080/10447318.2024.2353451.

[37] T. Alves, J. Natálio, J. Henriques-Calado, and S. Gama, "Incorporating personality in user interface design: A review", *Personality and Individual Differences*, vol. 155, p. 109 709, Mar. 1, 2020, ISSN: 0191-8869. DOI: https://doi.org/10.1016/j.paid.2019.109709. Accessed: Jul. 1, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S019188691930649X.

[38] S. Abrahão, E. Insfran, and A. Sluÿters, "Model-based intelligent user interface adaptation: Challenges and future directions", *Softw Syst Model*, vol. 20, Jun. 29, 2021. Accessed: Jan. 7, 2025. [Online]. Available: https://doi.org/10.1007/s10270-021-00909-7.

[39] M. H. Miraz, M. Ali, and P. S. Excell, "Adaptive user interfaces and universal usability through plasticity of user interface design", *Computer Science Review*, vol. 40, p. 100 363, Jan. 13, 2021, ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2021.100363. Accessed: Jan. 7, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013721000034.

[40] Z. Stefanidi, G. Margetis, S. Ntoa, and G. Papagiannakis, "Real-time adaptation of context-aware intelligent user interfaces, for enhanced situational awareness", *IEEE Access*, vol. 10, pp. 23 367–23 393, Feb. 18, 2022. DOI: 10.1109/ACCESS.2022.3152743. Accessed: Mar. 19, 2026.

[41] M. Alipour, É. Céret, and S. Dupuy-Chessa, "A framework for user interface adaptation to emotions and their temporal aspects", *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. EICS, Jun. 19, 2023. DOI: 10.1145/3593238. Accessed: Mar. 19, 2026. [Online]. Available: https://doi.org/10.1145/3593238.

[42] S. Duan, Z. Wang, S. Wang, M. Chen, and R. Zhang, "Emotion-aware interaction design in intelligent user interface using multi-modal deep learning", in *2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*, IEEE Xplore, Nov. 8, 2024. DOI: 10.1109/ISCEIC63613.2024.10810240. Accessed: Jan. 8, 2025.

[43] X. Zhan, Y. Xu, and Y. Liu, "Personalized ui layout generation using deep learning: An adaptive interface design approach for enhanced user experience",

Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023, vol. 6, no. 1, Dec. 4, 2024. DOI: 10.60087/jaigs.v6i1.270. Accessed: Jan. 8, 2025. [Online]. Available: https://ojs.boulibrary.com/index.php/JAIGS/article/view/270.

[44] P. Nama, "Ai-powered mobile applications: Revolutionizing user interaction through intelligent features and context-aware services", *Journal of Emerging Technologies and Innovative Research*, vol. 10, g611–g620, Jan. 2023. Accessed: Jan. 8, 2025. [Online]. Available: https://www.researchgate.net/profile/Prathyusha-Nama/publication/385207252_AI-Powered_Mobile_Applications_Revolutionizing_User_Interaction_Through_Intelligent_Features_and_Context-Aware_Services/links/671a62bcedbc012ea13d0a09/AI-Powered-Mobile-Applications-Revolutionizing-User-Interaction-Through-Intelligent-Features-and-Context-Aware-Services.pdf.

[45] G. Pei, H. Li, Y. Lu, Y. Wang, S. Hua, and T. Li, "Affective computing: Recent advances, challenges, and future trends", *Intelligent Computing*, vol. 3, p. 0076, 2024-01-05. DOI: 10.34133/icomputing.0076. eprint: https://spj.science.org/doi/pdf/10.34133/icomputing.0076. Accessed: Mar. 19, 2026. [Online]. Available: https://spj.science.org/doi/abs/10.34133/icomputing.0076.

[46] Y. Wang et al., "A systematic review on affective computing: Emotion models, databases, and recent advances", *Information Fusion*, vol. 83-84, pp. 19–52, Jul. 2022, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2022.03.009. Accessed: Mar. 19, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522000367.

[47] K. Cortiñas-Lorenzo and G. Lacey, "Toward explainable affective computing: A review", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13 101–13 121, May 23, 2023. DOI: 10.1109/TNNLS.2023.3270027. Accessed: Mar. 19, 2026.

[48] V. Nanjappan, H.-N. Liang, W. Wang, and K. L. Man, "Chapter 1 - big data: A classification of acquisition and generation methods", in *Big Data Analytics for Sensor-Network Collected Intelligence*, ser. Intelligent Data-Centric Systems, Academic Press, 2017, ISBN: 978-0-12-809393-1. DOI: https://doi.org/10.1016/B978-0-12-809393-1.00001-5. Accessed: Jul. 1, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128093931000015.

[49] Y. Sekhavat, M. Sisi, and S. Roohi, "Affective interaction: Using emotions as a user interface in games", *Multimed Tools Appl*, vol. 80, Sep. 29, 2020. DOI: https://doi.org/10.1007/s11042-020-10006-4. Accessed: Feb. 10, 2025. [Online]. Available: https://doi.org/10.1007/s11042-020-10006-4.

[50] X. Zhao, L. Wang, and Y. e. a. Zhang, "A review of convolutional neural networks in computer vision", *Artif Intell Rev*, vol. 57,99, Mar. 23, 2024. Accessed: Mar. 20, 2026. [Online]. Available: https://doi.org/10.1007/s10462-024-10721-6.

[51] O. A. Montesinos López, A. Montesinos López, and J. Crossa. "Convolutional neural networks". [Online]. Available: https://doi.org/10.1007/978-3-030-89010-0_13.

[52] M. Krichen, "Convolutional neural networks: A survey", *Computers*, vol. 12, no. 8, Jul. 28, 2023, ISSN: 2073-431X. DOI: 10.3390/computers12080151. Accessed: Mar. 20, 2026. [Online]. Available: https://www.mdpi.com/2073-431X/12/8/151.

[53] Y. Li, "Research and application of deep learning in image recognition", in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, Mar. 1, 2022, pp. 994–999. DOI: 10.1109/ICPECA53709.2022.9718847. Accessed: Mar. 20, 2026.

[54] P. Salamon et al., "Classification confidence in exploratory learning: A user's guide", *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 803–829, Jul. 21, 2023, ISSN: 2504-4990. DOI: 10.3390/make5030043. Accessed: Mar. 20, 2026. [Online]. Available: https://www.mdpi.com/2504-4990/5/3/43.

[55] R. Rosales, P. Popov, and M. Paulitsch, *Evaluation of confidence-based ensembling in deep learning image classification*, Mar. 3, 2023. arXiv: 2303.03185 [cs.CV]. Accessed: Mar. 20, 2026. [Online]. Available: https://arxiv.org/abs/2303.03185.

[56] A. Hekler, L. Kuhn, and F. Buettner. "Beyond overconfidence: Foundation models redefine calibration in deep neural networks". arXiv: 2506.09593 [cs.LG], Accessed: Mar. 20, 2026. [Online]. Available: https://arxiv.org/abs/2506.09593.

[57] D. Corradini, A. Zampieri, M. Pasqua, E. Viglianisi, M. Dallago, and M. Ceccato, "Automated black-box testing of nominal and error scenarios in restful apis", *Software Testing, Verification and Reliability*, vol. 32, no. 5, e1808, Jan. 23, 2022. DOI: https://doi.org/10.1002/stvr.1808. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stvr.1808. Accessed: Mar. 20, 2026. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/stvr.1808.

[58] W. von Eschenbach, "Transparency and the black box problem: Why we do not trust ai", *Philos. Technol.*, vol. 35, Sep. 1, 2021, ISSN: 1607–1622. Accessed: Mar. 20, 2026. [Online]. Available: https://doi.org/10.1007/s13347-021-00477-0.

[59] X. Guo, Y. Zhang, and S. e. a. Lu, "Facial expression recognition: A review", *Multimed Tools Appl*, vol. 83, Aug. 17, 2023, ISSN: 23689–23735. Accessed:

Feb. 6, 2025. [Online]. Available: https://doi.org/10.1007/s11042-023-15982-x.

[60] T. Kopalidis, V. Solachidis, N. Vretos, and P. Daras, "Advances in facial expression recognition: A survey of methods, benchmarks, models, and datasets", *Information*, vol. 15, no. 3, Feb. 28, 2024, ISSN: 2078-2489. DOI: 10.3390/info15030135. Accessed: Feb. 6, 2025. [Online]. Available: https://www.mdpi.com/2078-2489/15/3/135.

[61] J. Lee and C. Kim, "A structure of basic emotions: A review of basic emotion theories using an emotionally fine-tuned language model", *Proceedings of the Annual Meeting of the Cognitive Science Society, 45*, 2023. Accessed: Mar. 20, 2026. [Online]. Available: https://escholarship.org/uc/item/2zd4f4dk.

[62] M. Karnati, A. Seal, D. Bhattacharjee, A. Yazidi, and O. Krejcar, "Understanding deep learning techniques for recognition of human emotions using facial expressions: A comprehensive survey", *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–31, Feb. 9, 2023. DOI: 10.1109/TIM.2023.3243661. Accessed: Mar. 20, 2026.

[63] Z. Song, "Facial expression emotion recognition model integrating philosophy and machine learning theory", *Frontiers in Psychology*, vol. 12, Sep. 27, 2021, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.759485. Accessed: Feb. 6, 2025. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2021.759485.

[64] S.-J. Park, B.-G. Kim, and N. Chilamkurti, "A robust facial expression recognition algorithm based on multi-rate feature fusion scheme", *Sensors*, vol. 21, no. 21, Oct. 14, 2021, ISSN: 1424-8220. Accessed: Feb. 6, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/21/21/6954.

[65] M. Mohana and P. Subashini, "Facial expression recognition using machine learning and deep learning techniques: A systematic review", *SN COMPUT. SCI.*, vol. 5, no. 432, Apr. 13, 2024. Accessed: Mar. 22, 2026. [Online]. Available: https://doi.org/10.1007/s42979-024-02792-7.

[66] F. Z. Canal et al., "A survey on facial emotion recognition techniques: A state-of-the-art literature review", *Information Sciences*, vol. 582, pp. 593–617, Oct. 2, 2021, ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2021.10.005. Accessed: Feb. 6, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521010136.

[67] D. Gera, S. Balasubramanian, and A. Jami, "Cern: Compact facial expression recognition net", *Pattern Recognition Letters*, vol. 155, pp. 9–18, Jan. 21, 2022, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2022.01.013. Accessed: Feb. 6, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865522000204.

[68] M. Kaur and M. Kumar, "Facial emotion recognition: A comprehensive review", *Expert Systems*, vol. 41, no. 10, e13670, Jun. 26, 2024. DOI: https://doi.org/10.1111/exsy.13670. Accessed: Feb. 10, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13670.

[69] X. Li, Z. Shi, J. Chen, and Y. Liu, "Realizing emotional interactions to learn user experience and guide energy optimization for mobile architectures", in *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct. 5, 2022, pp. 868–884. DOI: 10.1109/MICRO56248.2022.00064. Accessed: Feb. 10, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9923805.

[70] Y. Chen, J. Li, S. Shan, M. Wang, and R. Hong, "From static to dynamic: Adapting landmark-aware image models for facial expression recognition in

videos", *IEEE Transactions on Affective Computing*, vol. 16, no. 2, pp. 624–638, 2025. DOI: 10.1109/TAFFC.2024.3453443. Accessed: Mar. 22, 2026.

[71] N. O and Y. N., "Api latency and user experience: What aspects impact latency and what are the implications for company performance?", *Digitala Vetenskapliga Arkivet*, Oct. 3, 2022. Accessed: Mar. 22, 2026.

[72] N. Aikyn, A. Zhanegizov, and T. e. a. Aidarov, "Efficient facial expression recognition framework based on edge computing", *J Supercomput*, vol. 1935–1972, 2023-07-24. Accessed: Mar. 22, 2026.

[73] K. Cui, G. Zhang, F. Zhang, and S. U. Khan, "Facial expression recognition system on a distributed edge-cloud infrastructure", in *2022 IEEE Cloud Summit*, Dec. 13, 2022, pp. 51–56. DOI: 10.1109/CloudSummit54781.2022.00014. Accessed: Mar. 22, 2026.

[74] K. Academy. "Katex", Accessed: Feb. 25, 2026. [Online]. Available: https://katex.org/docs/api.

[75] D. Jonauskaite and C. Mohr, "Do we feel colours? a systematic review of 128 years of psychological research linking colours and emotions", *Psychon Bull Rev*, vol. 32, 2025-01-13, ISSN: 1457–1486. Accessed: Apr. 21, 2026. [Online]. Available: https://doi.org/10.3758/s13423-024-02615-z.

[76] I. Amazon Web Services. "Amazon rekognition resources", Accessed: Feb. 25, 2026. [Online]. Available: https://docs.aws.amazon.com/pdfs/rekognition/latest/dg/rekognition-dg.pdf.

[77] G. Cloud. "Cloud vision api documentation and features", Accessed: Feb. 25, 2026. [Online]. Available: https://docs.cloud.google.com/vision/docs/features-list.

[78] F. of Life Institute. "Eu artificial intelligence act", Accessed: Mar. 23, 2026. [Online]. Available: https://artificialintelligenceact.eu/high-level-summary/.

[79] A. Fola-Rose, E. Solomon, K. Bryant, and A. Woubie, "A systematic review of facial recognition methods: Advancements, applications, and ethical dilemmas", in *2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, Aug. 2024, pp. 314–319. DOI: 10.1109/IRI62200.2024.00070. Accessed: Mar. 23, 2026.

[80] L. Hogenhout and R. Wangmo, *Protecting persona biometric data: The case of facial privacy*, 2025-10-03. arXiv: 2510.03035 [cs.CR]. Accessed: Apr. 22, 2026. [Online]. Available: https://arxiv.org/abs/2510.03035.

[81] F. Qazi, "Application programming interface (api) security in cloud applications", *EAI Endorsed Transactions on Cloud Systems*, vol. 7(23), 2023. Accessed: Apr. 22, 2026. [Online]. Available: https://doi.org/10.4108/eetcs.v7i23.3011.

[82] T. Yang and J. e. a. Zhang Y.and Sun, "Privacy enhanced cloud-based facial recognition", *Neural Process Lett*, vol. 54, 2022, ISSN: 2717–2725. Accessed: Apr. 22, 2026. [Online]. Available: https://doi.org/10.1007/s11063-021-10477-y.

[83] C. Chen, M. Sun, X. Gong, Y. Chen, and Q. Wang, *A survey on facial image privacy preservation in cloud-based services*, 2025. arXiv: 2501.08665 [cs.CV]. Accessed: Apr. 22, 2026. [Online]. Available: https://arxiv.org/abs/2501.08665.

[84] A. Katirai, "Ethical considerations in emotion recognition technologies: A review of the literature", *AI Ethics*, vol. 4, Jun. 20, 2023, ISSN: 927–948. Accessed:

Mar. 23, 2026. [Online]. Available: https://doi.org/10.1007/s43681-023-00307-3.

[85] D. Barker, M. K. R. Tippireddy, A. Farhan, and B. Ahmed, "Ethical considerations in emotion recognition research", *Psychology International*, vol. 7, no. 2, 2025, ISSN: 2813-9844. DOI: 10.3390/psycholint7020043. Accessed: Mar. 23, 2026. [Online]. Available: https://www.mdpi.com/2813-9844/7/2/43.

[86] I. Amazon Web Services. "Gdpr", Accessed: Mar. 23, 2026. [Online]. Available: https://aws.amazon.com/compliance/gdpr-center/.

[87] G. Cloud. "Privacy notice", Accessed: Mar. 4, 2026. [Online]. Available: https://cloud.google.com/terms/cloud-privacy-notice.

[88] G. Cloud. "Gdpr and google cloud", Accessed: Mar. 23, 2026. [Online]. Available: https://cloud.google.com/privacy/gdpr?hl=en.

[89] B. Sumer, "When do the images of biometric characteristics qualify as special categories of data under the gdpr?: A systemic approach to biometric data processing", in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, Sep. 27, 2022, pp. 1–6. DOI: 10.1109/BIOSIG55365.2022.9897034. Accessed: Apr. 22, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9897034.

Appendix A Post-Study Survey

The participants of the experiment filled out the following survey immediately after completing the experiment tasks. Responses were collected via Google Forms. Likert-scale questions used a 1–7 scale. Questions 22–26 allowed multiple selections from a fixed list. Questions 28–31 were conditional and presented only to participants who indicated they had noticed a change in the user interface (Q27).

Part 1: Background

Q1. Participant Code

[Free text]

Q2. How would you rate your mathematical abilities in general?

Passable ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Expert*

1 2 3 4 5 6 7

Part 2: Stage 1 — Additions and Subtractions

Q3. How stressed did you feel when completing the first section of calculations (additions, subtractions)?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q4. How confident did you feel during this stage?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q5. Select all the boxes that match your emotions during Stage 1 (additions and subtractions).

☐ Happy	☐ Calm	☐ Focused	☐ Neutral
☐ Stressed	☐ Frustrated	☐ Confused	☐ Anxious
☐ Bored	☐ Sad	☐ Irritated	☐ Surprised

Part 3: Stage 2 — Multiplication and Fractions

Q6. How stressed did you feel when completing the second section of calculations (multiplication, fractions)?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q7. How would you describe how you felt in this stage compared to the previous stage?

Less stressed ☐ ☐ ☐ ☐ ☐ ☐ ☐ *More stressed*

1 2 3 4 5 6 7

Q8. How confident did you feel during this stage?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q9. Select all the boxes that match your emotions during Stage 2 (multiplication, fractions).

☐ Happy	☐ Calm	☐ Focused	☐ Neutral
☐ Stressed	☐ Frustrated	☐ Confused	☐ Anxious
☐ Bored	☐ Sad	☐ Irritated	☐ Surprised

Part 4: Stage 3 — Simple Differentials

Q10. How stressed did you feel when completing the third section of calculations (simple differentials)?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q11. How would you describe how you felt in this stage compared to the previous stage?

Less stressed ☐ ☐ ☐ ☐ ☐ ☐ ☐ *More stressed*

1 2 3 4 5 6 7

Q12. How confident did you feel during this stage?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q13. Select all the boxes that match your emotions during Stage 3 (simple differentials).

☐ Happy	☐ Calm	☐ Focused	☐ Neutral
☐ Stressed	☐ Frustrated	☐ Confused	☐ Anxious
☐ Bored	☐ Sad	☐ Irritated	☐ Surprised

Part 5: Stage 4 — Difficult Differentials

Q14. How stressed did you feel when completing the fourth section of calculations (difficult differentials)?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q15. How would you describe how you felt in this stage compared to the previous stage?

Less stressed ☐ ☐ ☐ ☐ ☐ ☐ ☐ *More stressed*

1 2 3 4 5 6 7

Q16. How confident did you feel during this stage?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q17. Select all the boxes that match your emotions during Stage 4 (difficult differentials).

☐ Happy	☐ Calm	☐ Focused	☐ Neutral
☐ Stressed	☐ Frustrated	☐ Confused	☐ Anxious
☐ Bored	☐ Sad	☐ Irritated	☐ Surprised

Part 6: Stage 5 — Complex Numbers and Differentiations

Q18. How stressed did you feel when completing the final section of calculations (complex numbers, complex differentiations)?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q19. How would you describe how you felt in this stage compared to the previous stage?

Less stressed ☐ ☐ ☐ ☐ ☐ ☐ ☐ *More stressed*

1 2 3 4 5 6 7

Q20. How confident did you feel during this stage?

Not at all ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Extremely*

1 2 3 4 5 6 7

Q21. Select all the boxes that match your emotions during Stage 5 (complex numbers, complex differentiations).

☐ Happy ☐ Calm ☐ Focused ☐ Neutral

☐ Stressed ☐ Frustrated ☐ Confused ☐ Anxious

☐ Bored ☐ Sad ☐ Irritated ☐ Surprised

Part 7: Overall Task Experience

Q22. Which stage did you think was the most difficult?

☐ Stage 1 ☐ Stage 2 ☐ Stage 3 ☐ Stage 4 ☐ Stage 5

Q23. The difficulty increased gradually during the stages.

Strongly disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Strongly agree*

1 2 3 4 5 6 7

Q24. I felt time-pressured during the exercises.

Strongly disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Strongly agree*

1 2 3 4 5 6 7

Q25. I was aware of my emotions while completing the exercises.

Strongly disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ *Strongly agree*

1 2 3 4 5 6 7

Part 8: Adaptive Interface Experience

Q26. Did you notice a change in the user interface during the test?

☐ Yes ☐ No

Questions Q27–Q30 were presented only to participants who answered Yes to Q26.

Q27. If yes, did you feel more supported when the user interface changed?

☐ Yes ☐ Somewhat ☐ No

Q28. If yes, did you feel more comfortable completing the exercises when the user interface changed?

☐ Yes ☐ Somewhat ☐ No

Q29. If yes, did the change help you feel less frustrated?

☐ Yes ☐ Somewhat ☐ No

Q30. If yes, did the change in the user interface feel distracting?

☐ Yes ☐ Somewhat ☐ No

Part 9: Open Feedback

Q31. What affected your emotions and experience positively the most?

[Open text response]

Q32. What affected your emotions and experience negatively the most?

[Open text response]

Appendix B Math Task Question Bank

The following questions were included in the tasks the participants of the experiment had to complete. Questions were organized into five stages of increasing difficulty. Each stage had a timer. Answers were entered as plain text and parsed by the system.

Table B.1: Stage overview

Stage	**Topic**	**Questions**	**Time limit**
1	Additions and subtractions	5	1 min 30 sec
2	Multiplication and fractions	5	2 min
3	Simple differentials	5	2 min
4	Difficult differentials	5	3 min
5	Complex numbers and integrations	5	3 min

Stage 1 — Additions and Subtractions

Time limit: 1 minute 30 seconds. *Answer type: integer.*

Q1. $47 + 28 =$ *Answer:* 75

Q2. $86 - 39 =$ *Answer:* 47

Q3. $284 + 367 =$ *Answer:* 651

Q4. $915 - 428 =$ *Answer:* 487

Q5. $640 + 235 - 118 =$ *Answer:* 757

Stage 2 — Multiplication and Fractions

Time limit: 2 minutes. *Answer types: integer, fraction (reduced form), or decimal.*

Q1. $72 \cdot 6 =$ *Answer:* 432

Q2. $125 \cdot 24 =$ *Answer:* 3000

Q3. $\left(\frac{7}{8}\right)$ of $1600 =$ *Answer:* 1400

Q4. $\frac{3}{5} + \frac{1}{10} =$ (reduced fraction) *Answer:* $\frac{7}{10}$

Q5. $\dfrac{25}{\sqrt{25}} =$ *Answer:* 5

Stage 3 — Simple Differentials

Time limit: 2 minutes. Answer types: derivative expression or integer. Equivalent notations were accepted (e.g. `e^x` *and* `exp(x)`*).*

Q1. Differentiate $f(x) = 4x^3 - 7x + 2$. Enter $f'(x)$. *Answer:* $12x^2 - 7$

Q2. Differentiate $f(x) = (x^2 + 1)(x - 3)$. Enter $f'(x)$ expanded. *Answer:* $3x^2 - 6x + 1$

Q3. Differentiate $f(x) = e^x + \ln(x)$. Enter $f'(x)$. *Answer:* $e^x + \dfrac{1}{x}$

Q4. If $f(x) = x^3 - 6x^2 + 9x$, compute $f'(3) =$ *Answer:* 0

Q5. If $f(x) = \sin(x) + x^2$, compute $f'(0) =$ *Answer:* 1

Stage 4 — Difficult Differentials

Time limit: 3 minutes. Answer types: derivative expression or integer. Participants were required to apply the product rule, quotient rule, and chain rule where applicable.

Q1. Differentiate $f(x) = x^2 \ln(x)$. Enter $f'(x)$. *Answer:* $2x \ln(x) + x$

Q2. Differentiate $f(x) = \dfrac{3x+2}{x^2}$. Enter $f'(x)$ in simplest form. *Answer:* $-\dfrac{3x+4}{x^3}$

Q3. Differentiate $f(x) = \sin(x)\, e^x$. Enter $f'(x)$. *Answer:* $e^x(\sin(x) + \cos(x))$

Q4. Differentiate $f(x) = \ln(1+x^2)$. Enter $f'(x)$. *Answer:* $\dfrac{2x}{1+x^2}$

Q5. If $f(x) = \cos(x)$, compute $f'(0) =$ *Answer:* 0

Stage 5 — Complex Numbers and Integrations

Time limit: 3 minutes. Answer types: decimal (1 d.p.), integer, or complex number in $a + bi$ form.

Q1. Approximate $\sqrt{137}$ to 1 decimal place = *Answer:* 11.7

Q2. Approximate $\ln(7)$ to 1 decimal place = *Answer:* 1.9

Q3. Compute $\displaystyle\int_0^2 (2x+3)\, dx =$ *Answer:* 10

Q4. Multiply $(3-4i)(-2+5i) =$ (enter in $a+bi$ form) *Answer:* $14 + 23i$

Q5. Solve $z^2 + 4z + 13 = 0$ (enter ONE root in $a+bi$ form)

Accepted answers: $-2+3i$ or $-2-3i$

All logic, parsing and normalization can be found in the thesis repository `https://github.com/sarakeskilohko/fer-adaptive-ui`