REFEREE: A SELF-REFLECTION TOOL FOR GAMERS

JEROEN RITMEESTER

UNIVERSITY
OF TWENTE.

BSc Creative Technology
Supervisor: Ansgar Fehnker
Critical observers: dr. Khiet Truong, dr. Alma Schaafstal
Faculty EEMCS
University of Twente

7 July 2020

ABSTRACT

Facial Expression Recognition has a great amount of potential, but has not gained the wide recognition of the public yet, like face recognition or FaceID. Introducing potential consumers as well as developers to this technology can open up many doors and innovative concepts in the field of Human Media Interaction. Making use of of convolutional neural networks, optionally aided by more physiological signals, emotion recognition can help in branches like psychology, security, and healthcare. In this graduation project thesis, a prototype for a fully functional interface is presented that focuses on emotive self-reflection in the video games industry. The prototype is evaluated and future improvements are discussed, with the intention that this prototype is elaborated upon in the future.

PREFACE

This thesis is the conclusion of my bachelor Creative Technology at the University of Twente. The resulting prototype was made with the complete intention of further elaboration by future graduation project students. I have been engaged in research and development for the purposes of this project since February 2020, and concluded in July 2020.

This project has allowed me to get a first look into the field of Affective Computing, combined with Artificial Intelligence, making use of convolutional neural networks. Although this is by no means an easy subject to step into for the first time, good supervision, feedback, and support from friends and family has allowed me to keep up the energy needed to complete this project.

I would like to thank my supervisor Dr. Ansgar Fehnker for the half-year of enthusiasm, support, and great spirits, as well as critical observers Dr. Khiet Truong and Dr. Alma Schaafstal for additonal valuable feedback and encouragement. Finally a special thanks to my parents for supporting me throughout all these years, giving me the opportunity to eventually be able to write this thesis and obtain my bachelor's degree, doing something I have greatly enjoyed.

Jeroen Ritmeester
Enschede, July 7th 2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In this graduation project the goal is to explore and develop a prototype for a mostly autonomous human-computer interaction system that measures the emotive state in the user while interacting with video / computer games. The system will attempt to measure the user's emotions using facial expression recognition (FER), a type of system that is comprised of facial feature detection algorithms alongside an autonomous emotion classifier. Together, the system can analyse the emotive behaviour of video gamers. In doing so, the measurements should amount to a timeline of the user's emotions on the full duration of a measurement session. This timeline, along with other possible significant findings, should be presented to the user to give them a reflection of their emotional changes during video games.

1.1 MOTIVATION

Nowadays people are constantly surrounded by electronic devices, among which are smartphones, laptops, and desktop personal computers, all of which can be used to play video games. For many years now there have been concerns and questions regarding whether or not video games have any significant effect on people's behavioural patterns, often focusing on aggression specifically. Although a number of research papers have been written on this, little research has been done on self-reflection emotion tracking tools that could help people show their behaviour in an attempt to moderate this alleged increase in aggression.

According to Statista [1], the number of people that play video games has grown from 1.8 billion people to 2.6 billion between 2014 and 2020. According to the predictions of the Global Games Market Report [2] reports the 2.5 billion gamers in 2019 have spent 152.1 billion US dollars, giving an idea of the scale of the video game industry. Since this is a roughly a third of the global population, it is important that people have access to a means of self-reflection regarding their behaviour when playing video games. Especially younger people will likely spend more time per day on average playing video games, and their mental, physical, and social well-being will inadvertently be influenced by their behaviour while gaming.

1.2 OBJECTIVES

The main objective is therefore to create and implement a prototypical system architecture which has a number of goals. Firstly, the system has to be fast enough in measuring the users' facial expressions in real-time while also running other applications like video games. This means the system can provide the user with information about themselves fluently. As this will require some form of machine learning, note that the goal of this project is not to develop, train and test a custom classifier. Any classification algorithm that will be used in this thesis will have been pre-trained,

as the time and resources to make a custom one are not available at the time of this project. Secondly, the measurements of the emotive states of the user have to be presented to the user in a meaningful way. If, for example, the system is used to measure how much aggressive feelings there are during an competitive online multiplayer game, the user should be presented with a visualisation of that emotion, displaying the measured points in time where the system could detect "angry" as a facial expression.

Finally, the supervisor of this project has requested that the code, which will be written in the Python 3 language, should be of such quality that it can be used to teach Creative Technology students. This means that the structure, encapsulation, and readability have to be well done and the system should be readable to people with little or no experience in programming in Python. Furthermore, this project aims the potential of FER in an application that is usable in an everyday setting. Facial recognition and related technologies are becoming commonplace themselves nowadays. Keeping in mind the objectives mentioned above, this project is an opportunity to provide students with source code from which they can learn about this technology, as well as provide non-students with a low-threshold application that gives them an idea of how such autonomous systems work.

1.3 CHALLENGES

In order to meet these objectives, there are several things that need to be considered. First and foremost, a suitable classification algorithm is required to categorise the frames that come from video feed that contains the user's face. The capabilities of the system will be designed around the functionality that the classifier provides. To deliver understandable and clean source code at the end of this project, no libraries that are not available for installation through pip will be used unless absolutely necessary. In doing so, all libraries needed for execution of the system can be installed with one simple command, removing the many possible errors that come with compiling third-party libraries manually. Additionally, at the time of writing, the SARS-CoV-2 pandemic is ongoing, restricting in-person usability testing severely. This means that whatever prototype is made, will have to mostly, if not completely digital to allow for any type of sharing with other people for testing and feedback. Needless to say, this will severely impact the design, although in what way is yet to be seen.

1.4 RESEARCH QUESTIONS

The most important and hence the main design question in this thesis is as follows:

How can the facial expression recognition be integrated into a system that allows creating awareness of potential of this technology?

Here, awareness has a double meaning: on the one hand, it points towards the knowledge of all people about this technology and its everyday impact, while on the other hand it represents the potential for the students when they use FER effectively. Because this question difficult to answer as a whole, the following sub-questions are posed:

- *How can a tool be designed to provide emotive self-reflection to people playing video games?*
- *How can that tool be implemented so that it raises awareness of the technology's potential?*
- *To what extent is there a place for such a tool among video gamers?*

1.5 REPORT OUTLINE

The following chapter will each comprise a phase in the design process of the envisioned tool. The next chapter will discuss relevant background theory that should help with development, like the Creative Technology Design Method. Chapter 3 will discuss the design methods and techniques used during this project. Chapters 4 through 6 are each a phase in the Creative Technology Design Method. More specifically, the fourth chapter discusses Ideation, which includes a brief discussion on this project's previous topic, conceptualisation of possible applications of the current topic, a stakeholder analysis and the first iteration's prototype. Chapter 5 will take the feedback from that iteration and further specify user needs and requirements. Finally, Chapter 6 explain the workings of the final prototype. After this, the system is evaluated to see to get high-level feedback from test users. The report ends with a conclusion and recommendations for future work.

2

BACKGROUND AND RELATED WORK

In this chapter a theoretical foundation will be laid out in to order to understand the workings of a system like this thesis is aiming to create. The underlying functionalities of some facial expression recognition systems are discussed to provide an understanding of how the end application will operate. Additionally, various categorisation of emotions are investigated, since different categorisations have different benefits and drawbacks.

2.1 AUTOMATED CODING OF FACIAL EMOTIVE RESPONSES

Facial expression recognition (FER) is a technology within the field of computer vision, which uses virtual markers to detect people's facially expressed emotions. It is used in psychological analysis, security, face expression synthesis, operator fatigue detection, and more. Because of its already established usage in the field of psychology, it may be a worthwhile endeavour to see to what extent FER can be used for analysing video game addiction, aggression, mood changes, et cetera.

Many tasks where manually annotating human emotions is highly impractical or impossible could be replaced by a robust enough FER system. Many systems make use of Ekman's facial action coding system (FACS) [3]. This is a "anatomically based system which breaks down facial expressions into individual components of muscle movement." The muscular activity during a facial expression can be categorised into action units (AUs). All AUs are mutually exclusive and therefore have no overlap. Each AU has an intensity between 0 (absent) and 5 (maximal) [4]. Combinations of these AUs amount to prototypic expressions of certain emotions. However, only by using the EMFACS (Emotional FACS) system or similar systems can face images be coded for emotion-specified expressions [5]. In FACS, Ekman states that there are seven basic expressions: happy, sad, angry, surprised, disgusted, fear and contempt. Additionally, there is a neutral state. The detection and application of the neutral expression varies per research.

To illustrate the workings of these system, EMFACS can code for the emotion "happy", if there is a sufficiently high score on AU6 and AU12, which are the *Orbicularis oculi* (cheek raiser) and *zygomaticus major* (lip corner puller) muscle groups, respectively (figure 1 [6]). If AU12 is sufficiently high, but AU6 is not, i.e. smiling without using your eyes, this is an indication of a fake smile, as opposed to a genuine smile or "Duchenne smile". This is how the system allows for a coding that can be used in classifiers to determine the intensity and genuineness of the emotions facially displayed by the user.

With the FACS system introduced, an important distinction should be noted between face detection, face recognition, and face authentication (also called identification or verification). Face detection analyses the input frames to detect the presence of a human face. In its most basic form, this is not different from regular object-class de-

Figure 1: AU6 (*Orbicularis oculi*) and AU12 (*zygomaticus major*) in action.

tection. Detection is used for auto-focusing systems on cameras, tracking people in a frame - not specific individuals - and sometimes used as a low-tier biometric input.

Face recognition, then, is an application of face detection. The goal of recognition is to determine which individual is in front of the camera, rather than detecting an arbitrary human face. Another key difference between the two is that, for recognition, the measured data has to be stored and compared against a database of personal facial features (1:n matching), whereas facial detection has no need of making use of stored data, unless specifically designed to do so. Applications of face recognition are found in law enforcement, advertising, social media and automated photo clustering. Although its uses are plentiful, it is generally not a robust system and can be fooled easily by showing the system a photograph or digital image instead of a human face, and it can have trouble with comparing images of poor quality with an input of high quality, or vice versa.

Finally, face authentication, as the name implies, aims to determine if the detected person is authentic. That is, to establish whether they are who they claim to be. Similar to online accounts making use of a username and password, authentication performs 1:1 matching to look for a match. In terms of safety, face authentication is a far superior to face recognition.

With that distinction, automated FER and their accompanying classifiers can aid particularly with facial recognition. Face authentication does not help in this coding, as emotions and security are not (currently) related. Face detection, however, can help in tracking the location of facial features for the system to code [7].

2.2 Classifier implementation and evaluation

Automated emotion classification is relevant in many scientific fields, including psychological assessment, such as identifying if depressed patients are at risk of reattempting suicide [8], measuring conscious or subconscious pain levels [9], and distinguishing between staged pain and genuine pain [10][11]. As Girard et al. [12] point out there has been very little follow-up work on these subjects, as the manual coding of the data for these exercises are intensely time consuming, on top of the time it takes to learn how to code it in the first place. The recommended time to learn how to manually code in the FACS system is between 50 and 100 hours [13]. Besides the effort it takes to code any length of footage, humans are bound to either make mistakes, or multiple encoders may disagree.

There are also some significant downsides to automating this process. What appear as insignificant environmental variables for humans when it comes to natural facial emotion recognition, such as pose, orientation of the face, lighting, facial differences (skin colour, complexion, gender, etc.) and many more, form a great challenge to classification algorithms. Therefore, automated classification is a subject that needs be understood well before being able to design and implement an automated recognition architecture.

Luckily, automated classification has been subject to a lot of scientific interest for many years now. Based on the FACS system, Cohn and Kanade constructed a database [14] and later expanded it into the CK+ database [15], while also attempting to devise a universally applicable metric to evaluate the classifiers that are being developed. For AU detection, they proposed using the area under the curve that is created by plotting the true positives against the false positives as the decision threshold varies. For emotion detection, they documented the result in a confusion matrix. Kukla et al. [16] also present a confusion matrix to show how well the classifiers have performed. Having a somewhat consistent evaluation metric across the field is important for many reasons, the largest of which being that there is no single consensus yet on what architecture such a classifier should have for optimal performance [15][17].

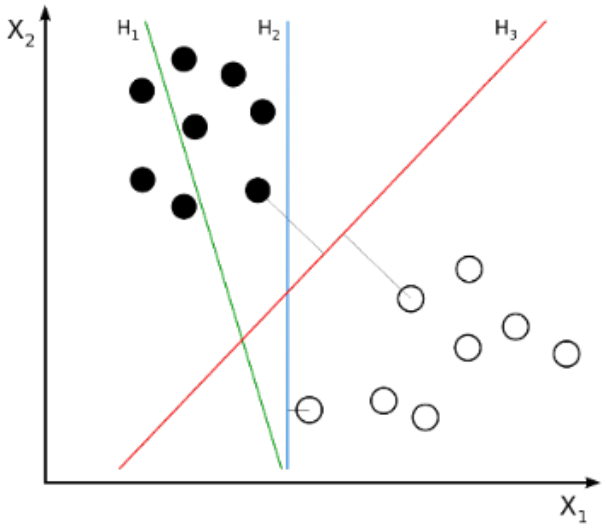

Figure 2: An SVM separating the entries into black and white dots, based on features X_1 and X_2. Line H_1 is incorrect. Line H_2 is correct now, but is prone to error. Line H_3 is preferable.

An often used approach is to employ support vector machines (SVM) [7][15]. An SVM attempts to separate all input features into two classes, based on the number of features n. This means it is a binary classifier, drawing a hyperplane (a separation between two groups in n dimensions) between the two classes. If $n = 2$, then the boundary between the two classes is just a line, separating a flat sheet into two parts. In figure 2, an example of a binary classification and possible boundaries is shown using $n = 2$ features, called X_1 and X_2. Boundary H_1 is incorrect, since it cuts through the black dots class, thus not providing a correct separation. H_2 does separate the two classes in this instance, but if a new dot is added only slightly to the right of

H_2, it would classify the new entry as white, which would intuitively be incorrect. Therefore, the optimal solution is a boundary where the distance to the nearest entries of each class is as large as possible. This is illustrated by boundary H_3.

A different approach is using convolutional neural networks (CNN) that output a likelihood or confidence level that indicates how confident the system is about a prediction. Kukla et al. [16] investigated a CNN, where each network attempts to find one of the seven basic emotions, and passes it on to the next network if it cannot find its assigned emotion. If none of the classifiers find any emotion, the output is the default state, namely neutral. This is schematically shown in figure 3.

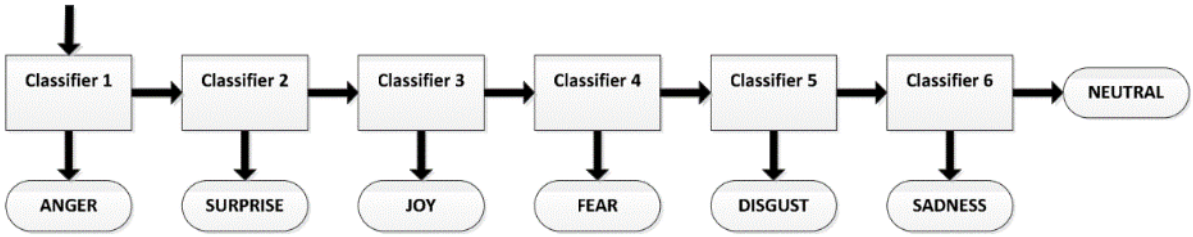

Figure 3: Chain of neural networks classifiers, where each classifier is trained to look for one specific emotion. The default state is neutral. Adapted from [16].

A significant problem, however, is that each classifier needs to be trained individually to recognise its assigned emotion. This requires a sufficiently large image set for each emotion, which is an time intensive task.

In many emotion tracking researches, FER is not the only source of information. Instead it is aided by the input of various sensors that track other actions of the user [18]. For example, in a scenario where the user is driving a car, the throttle, brake, and steering patterns may be included to make a prediction about the user's emotional state, either independently or in cooperation with the FER system. In many papers, EMG (electromyography) is used to measure muscular activity in the face to help the otherwise visual-only FACE coding for facial muscles. In other settings galvanic skin response ,or skin conduction, may be measured to test for arousal, which is a factor of a different type of emotional categorisation (see Section 2.3). Having more parameters to base a classification on generally ensures robustness and increases prediction accuracy [19].

2.3 MODELS FOR EMOTIVE STATES

Various models have been designed to describe emotions in a systemic way. Although some models are nothing alike, one is not necessarily less correct than another. A model's usefulness depends largely on the application of the model. This section will highlight a few models used in affective computing, which is the field that concerns "computing that relates to, arises from, or influences emotions" [20].

2.3.1 *Basic emotion theory*

Within this field, there are two major opposing theories with fundamental differences. The first is the theory of basic emotion [21], which says that different emotions arise from distinct neural systems, and thus do not arise from one universal emotional complex in the brain. Basic emotion theory is by far the oldest, originating in ancient Greece and China [22]. The current version of this theory started with Darwin [23]. It is versions of this model on which Ekman would eventually build the FACS system, presenting the basic emotions as the aforementioned set of seven: happy, sad, angry, surprise, disgust, fear, and contempt, as seen in figure 4.

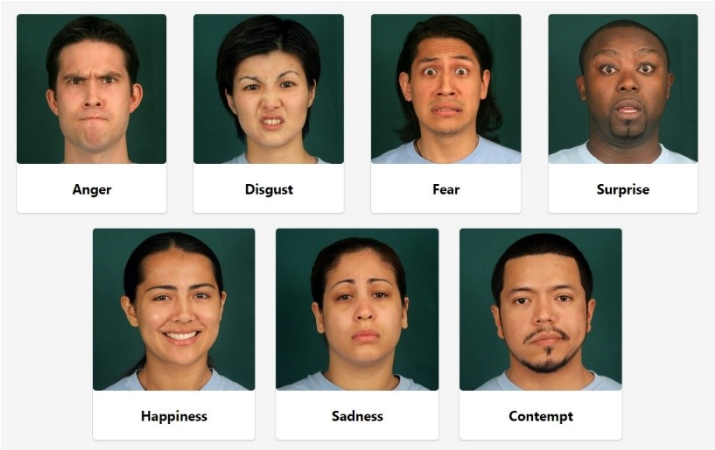

Figure 4: Expressions of Ekman's basic emotions

As this theory forms the basis of the FACS model, this system is highly relevant to this thesis. However, there is a wide array of models that subdivide Ekman's six basic emotions into more detailed ones. Robinson [24] subdivides them according to their degrees of complexity within the domain of emotions, a process that yields eleven pairs of positive and negative emotions (table 1).

2.3.2 *Dimensional theory*

The opposite theory, named the dimensional theory, has led to a large number of different dimensional models of its own. These models each have their own way of organising affective states, i.e. emotions, according to similar but slightly differently defined metrics. This class of models may be of useful, as assigning a two-dimensional valence-arousal score to images produces a much more informative description of an image than categorising it with a one-word emotion. There are a couple prominent models used in the field of psychology and affective computing. The most common one is called the circumplex model [21]. It says that there is a common and interconnected neurophysiological system that is responsible for all affective states, and thus all emotions. The circumplex model uses only two metrics: valence, which indicates how positive or negative is something, and arousal, which is the emotional intensity ranging from "calm" to "exciting" [25]. All affective states can be mapped to this two-dimensional valence-arousal space, as seen in figure 5 [26]. In this valence-

Kind of emotion	Positive emotions	Negative emotions
Emotions related to object properties	Curiosity, interest	Panic, alarm
	Desire, attraction	**Disgust**, aversion, revulsion
	Surprise, amusement	Indifference, familiarity, habituation
Future appraisal emotions	Hope	**Fear**
Event related emotions	Gratitude, thankfulness	**Anger**, rage
	Joy, elation, triumph, jubilation	**Sorrow**, grief
	Relief	Frustration, disappointment
Self-appraisal emotions	Pride in achievement, self-confidence, sociability	Embarrassment, shame, guilt, remorse
Social emotions	Generosity	Avarice, greed, miserliness, envy, jealousy
	Sympathy	Cruelty
Cathected emotions	Love	Hate

Table 1: Robinson's subdivison of the basic emotions.

arousal space, the horizontal axis indicates the level of arousal, where it increases in positivity to the right-hand side and increases in negativity on the left-hand side. The vertical axis indicates arousal, where it increases in arousal upwards and similarly decreases downwards. The two axes make for four quadrants that group similar emotions together into angry, happy, sad and relaxed (figure 6) [27].

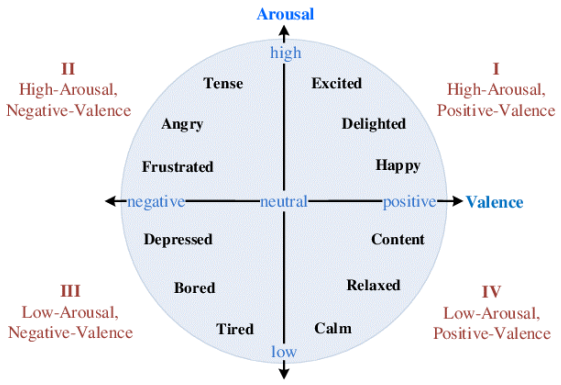

Figure 5: The two-dimensional plane of valence and arousal.

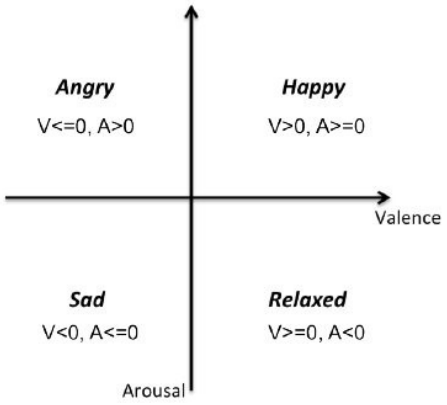

Figure 6: Valence-arousal plane simplified into quadrants.

2.4 EFFECT OF VIDEO GAMES ON EMOTIVE STATES

Video games can have a significant impact on the emotive state and emotional well-being of the people who play it, depending on several factors, such as the players' age, affinity with technology, and other interests. For many years now, a public concern persists regarding the health of people that engage for longer periods of time in playing video games, particularly younger people [28]. In fact, there has been found a positive correlation between video games containing violence and aggressive behaviour in the players [29]. However, equally evident is that other types of video games can have a positive effect on one's mental wellbeing. For example, puzzle video games – defined by Granic [30] as "games with minimal interfaces, short-term commitments, and a high degree of accessibility (e.g. Angry Birds, Bejewled)" – have a positive effect on players' moods, promotes relaxation, and even wards off anxiety [30]. As Bavelier [31] summarised concisely, "One can no more say what the effects of video games are, than one can say what the effects of food are." This ambiguity underlines the potential for a system than can track your emotions for you. Emergent patterns after longer use of such a system might show a trend towards feeling an emotion more often. This trend can then be enforced by active self-reflection.

2.5 RELATED WORK

A concrete example of a system using FER and automated classifiers that is in the same spirit as this thesis was carried out by McDuff et al. [17]. In this study, facial images were collected over the Internet on a large scale (totalling approximately two million facial video responses to 10,854 unique pieces of media content). From these responses, the facial region of interest (ROI) was extracted per frame and from this, histogram of oriented gradients (HOG) for each 32x32 pixel block. Finally, these HOGs were fed into a support vector machine classifier and post-processed. This yields a per-frame output of facially expressed emotion classes. The process is schematically shown in figure 7. They used their predictions for emotion-based video advertisement targeting. Other possibilities include video recommendation, content tagging and retrieval (indexing), and a couple more options.

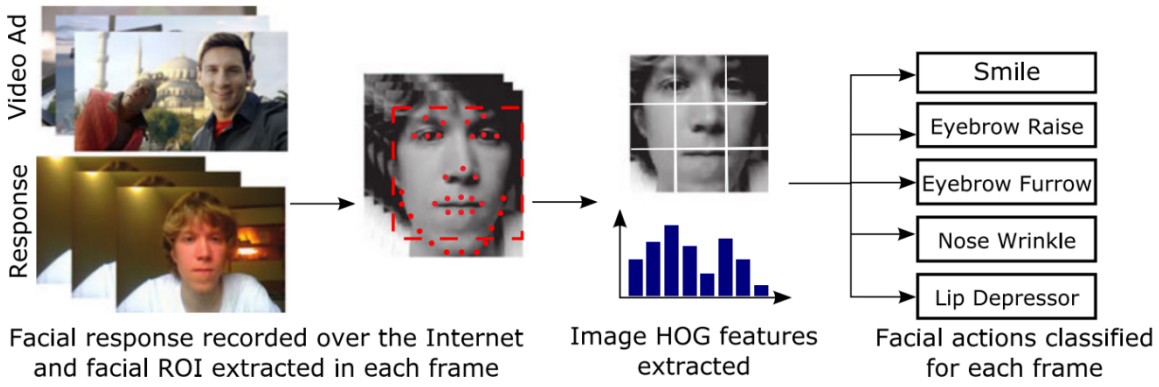

Figure 7: : Graphical representation of classification based on histograms of oriented gradients by McDuff et al. [17]

Training and utilising a custom classifier takes a great amount of work. McDuff used over 15,000,000 frames of human-coded frames, which is far beyond the capabilities of this thesis. The fact that all this is possible through the use of FER once again highlights the importance of user awareness of this technology.

A similar study by Zhao et al. [32] managed to design and implement an architecture that uses Haar-like features to measure facial expressions over time. Whereas active shape models are often used [15][33] as geometrical features to represent the appearance of the face, these are very sensitive to illumination, pose, and exaggerated expressions [34]. A system trying to find these features will use Haar kernels as seen in figure 8. The resulting Haar-like features are remarkably simple and effective [35] and are used to find specific parts of the face, as shown in figure 9 [36].

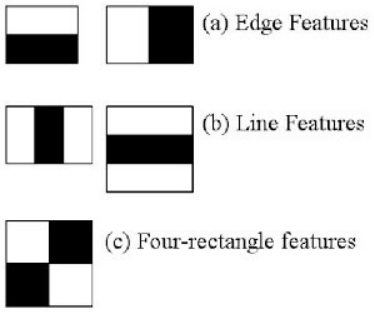

Figure 8: Haar kernels used to define regions of particular contrast.

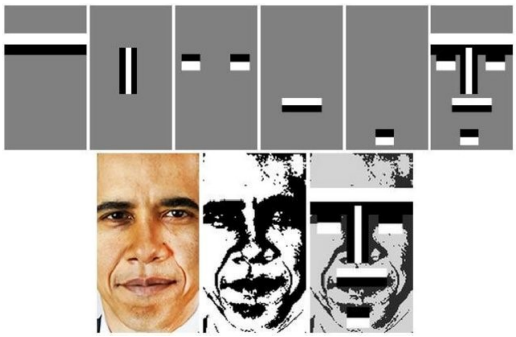

Figure 9: Detecting Haar-like features detected on a black-and-white image makes finding facial features easy and fast.

Finally, the features are used in a conditional random field (CRF), which is also a means of classification, similar to SVMs. The key difference with CRFs, or in this study's case hidden CRFs, is that it allows the context of a data point to be taken into account. This means that the correlation between separate Haar-like features is accounted for when classifying the emotion that is present on the user's face. The study then uses this classification in combination with temporal data to output a video recommendation for the user. In short, the user's own facial expression yields a video that the user would like to watch.

A third study by Joho et al. [37] designed a system that aims to create personalised recommendation system by automatically annotating the content of the video, again based of facial expression recognition. They accomplish this by continuously trying to detect the user's affective state through motion units as opposed to action units, which are then fed into a Naive Bayesian classifier. The goal of their approach is to filter out the highlight for individual users in order to recognise the content that they enjoyed most, and which can then be used as the input for a recommendation system. The previous subject of this thesis was similar to this. More on than can be found in Section 4.1.

3

METHODS AND TECHNIQUES

3.1 CREATIVE TECHNOLOGY DESIGN METHOD

In the bachelor Creative Technology, the Creative Technology Design Method [38] is used throughout its curriculum, and will serve as the backbone of the design process in this thesis. figure 10 depicts a schematic overview of the process. This methodology boasts an iterative approach between phases, allowing the outcomes of a later phase to be an input of a second iteration of an earlier phase. This is done in order to allow for multiple versions of the same concept, getting increasingly better and closer to the requirements and goals set at the start

The design process starts with a stakeholder analysis, followed by a pen-and-paper prototype created in the ideation phase. During this phase, concepts are formed based on the user requirements found in the stakeholder analysis. With the conceptual prototype, the ideas can be discussed with experts in order to make structural changes to the design. In the specification phase, these changes and other suggestions are evaluated and made more concrete, so they can be implemented in the realisation phase. Finally, the final prototype that results from the realisation phase is evaluated using test users, and their evaluation is documented and possible future improvements are discussed.

Note that the ideation phase normally starts by iterating on ways of creating some pre-known application or technology. In the case of this project, this application has not been defined yet. Instead, the starting point is the fact that FER should be implemented in a yet unknown application. Therefore, the ideation phase starts with a divergence-convergence cycle to create a suitable application first, and continuous normally from there on.

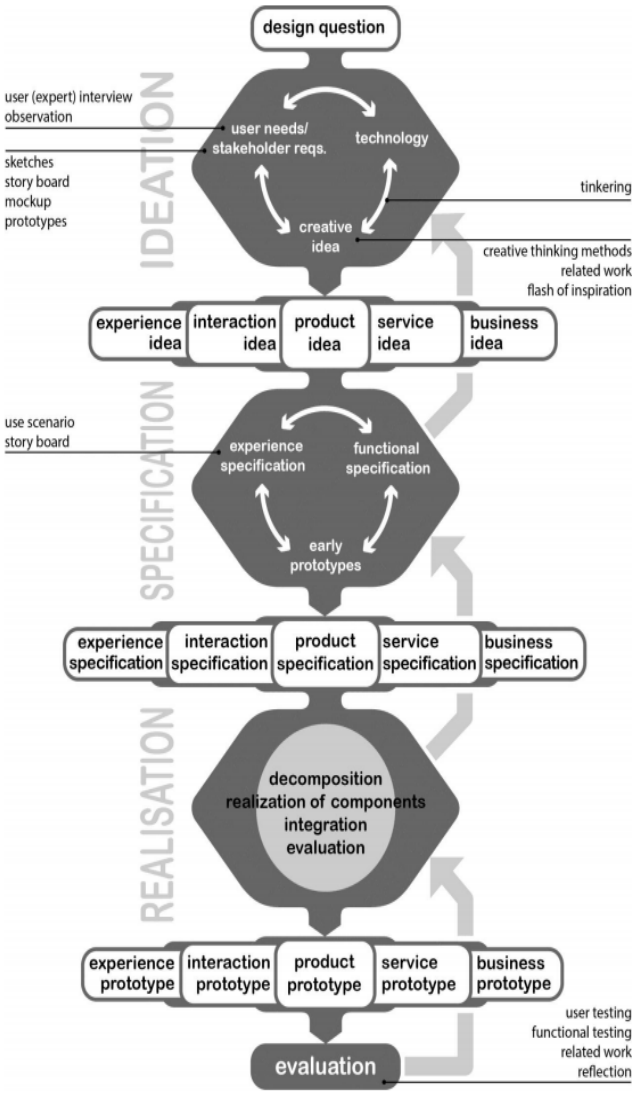

Figure 10: Creative Technology Design Process

3.2 NON-SPECIFIC SURVEYING

In order to get an insight in the behaviour of possible users, non-specific surveying is used to gather information, in the case of this project on emotional awareness, video game habits, and user input regarding possible applications for this system. This surveying is done fully anonymously over the Internet, to allow the user to voice their answers as freely as possible to ensure honest results. The results are processed by hand. The resulting information is not strictly necessary for the development of this system, but it does help to get a sense of people's game preferences and their emotional experiences in their favourite genres.

3.3 CO-CREATION SESSIONS AND USER INPUT

Throughout the design process, there will be multiple iterations of a prototype, starting with a mock-up "pen-and-paper" version of how the graphical user interface (GUI) should look like, followed by programmatical implementations. These versions will be shown to various people to gather ideas and informal feedback along the way. This ensures the product is something users will want to use. If the FER system is already in place, some people could also give an indication of the accuracy of the emotion classifier. At the end of the design process, random users that enjoy playing games will be asked to use the prototype when they are playing games in their own time. This way, the prototype can be tested in the environment that it is designed to be used.

4

IDEATION

In this chapter, the goal is to generate multiple ideas that can be elaborated upon and explored. Since the current topic was not the topic this thesis started out with, this section shows the ideation that eventually resulted in the current topic. Sections 4.1 through 4.3 show the process that led up to the selection of the current application. With the selection done, Section 4.4 and after discuss the ideation of the selected application.

4.1 INITIAL CONCEPT APPLICATIONS

Since emotional experience of users is an interesting aspect to measure in plenty of settings, this section ideates upon the application of such a system. Although many more concepts have been worked out, for sake of brevity, the five most promising ideas will be listed and a brief explanation will be provided for each of them. Some applications may prove to be of interest in multiple platforms, or require more than one platform to function. At the end of this chapter, in Section 4.6, the most promising idea will be selected and elaborated upon.

The concept "Creating awareness of the influence of autonomous facial recognition" described below, was pursued earlier. The goal then was to create an a system that tries to steer your emotion by showing you videos from the internet that elicit a certain emotion. Using FER, the system would measure your response and see how well it was succeeding in changing your behaviour to its preset goal. For example, if the system chose "Angry" as a pre-selected emotion, it would try to steer the user to feeling angry, by showing them videos with enraging content. The objective of this system was to show you how anyone can be manipulated by cleverly designed algorithms, especially if the usage of the collected data is not made fully transparent. This does occasionally happens with malicious intent. However, some of the world's largest technology companies, like Google and Amazon are already creating a guidelines to prevent such things from happening. Many other steps have already been taken to prevent privacy infringement and online fraud through law, especially in the European Union through the General Data Protection Regulation (GDPR), rendering that subject partially redundant.

- **Creating awareness of the influence of autonomous facial recognition** More and more people carry around handheld devices with cameras, and computing power necessary for facial (expression) recognition in public is becoming more ubiquitous. In some public areas, facial tracking is already being used by law enforcement to track people of interest. However, this technology could also be used malevolently. If used for marketing, not taking into account privacy law, web shop applications can learn more about your behavioural patterns when shopping than you would like to give away.

Advantages: This teaches the user about personal digital security. Could be used in educational setting as a teaching tool on security on the Internet.

Disadvantages: There is already a large amount of privacy concerns turned into law or guidelines. FER is currently not commonly used maliciously, and with this law making in progress, it will have less chance to become relevant, rendering the goal of this application irrelevant as well.

- **Automatic multimedia curation** A huge amount of on-demand content is being consumed worldwide, for example on Netflix, YouTube, Spotify, etc. Most of such platforms allow for a user rating, often in the form of "likes", or a numerical rating on a scale. Using FER, this could be automated, to further improve on the recommendation systems most of these platforms have in place.

  Advantages: Will be usable for almost everyone, allowing for a large amount of user data.

  Disadvantages: This is a very practical applications with no research question of its own. Only the usability and functionality are of interest as-is. Also, each application will need access to either an existing or self-made API with the platforms in questions. This may turn out to be very time consuming.

- **Aid when studying** While reading scientific papers, articles, mathematical explanations, or other sometimes difficult pieces of literature, FER can be employed to recognise and mark particularly difficult or troublesome sections. This can do two things: create a highlighted summary of sections that will need the user's attention later on for revision, and give the user insights in their emotional behaviour throughout reading or study session.

  Advantages: Will be usable for almost everyone, allowing for a large amount of user data. Particularly useful in a university setting, as there are plenty of students that will have more reading to do than most people outside of an educational setting.

  Disadvantages: Will require a lot of time to fine-tune the recognition system to result in useful highlighting of difficult parts. Also, it requires the user to always have the system running on a device capable of providing enough computational power while reading digital and non-digital texts. Finally, to allow for useful referencing, the system would have to be aware of what type texts are being read to avoid confusion after a long session in which multiple different texts were read.

- **Art object curation** FER can be used by museums or art galleries to gather the facially expressed opinions of visitors of specific objects. This is somewhat similar to "Automatic multimedia curation", but in this case it is used by another party than the people who are being measured.

  Advantages: Allows any type of museum or gallery to optimise their exhibits for their customers, similar to digital multimedia platforms.

  Disadvantages: Measuring many people in art galleries might be conflicting with privacy rules and regulations. Any measurements would likely require

to visitors' consent. On top of that, if the system is to be used for multiple art objects, multiple FER systems capable of providing enough computational power, or a convoluted network of cameras will be needed.

- **Self-reflection tool of behaviour during video games** As most singleplayer games are designed to elicit some emotion, and many multiplayer having a competitive nature, a wide array of emotions are prompted when playing video games. With the video games being ubiquitous nowadays in the life of many people of all ages, it is important for people to have a tool that allows them to see how their emotions change during games.

  Advantages: A large portion of university students engage in video games, allowing for a large group of possible users/testers. Furthermore, the system ties in to a large amount of research on the topic of behavioural change as the result of increased video game consumption.

  Disadvantages: This will require an elegant solution to prevent the user's system to overload or freeze when playing games, while also running an FER program.

4.2 General requirements

Thus far, the only few requirements have been posed: FER implementation has to be utilised in the project, and the code has to be clean and simple enough so that it could be used in the curriculum of BSc Creative Technology, as requested by this project's supervisor. More on this in the next section. FER and any other type of machine learning algorithm is most suitable on a desktop computer or heavy-duty cloud server that can manage a great number of calculations, preferably with access to a graphical processing unit (GPU) rather than only a central processing unit (CPU). As GPUs are specifically designed for a high number of parallel calculations, training, testing, and using any algorithm of this sort will be many times faster than on a CPU. In the next section, candidate applications will run on various devices, like desktop personal computer, mobile phones, and web browsers.

In the case of a desktop PC application, the program can be completely self-contained: the program can run on the machine in the foreground or background depending on its application, provided the owner of the system has a webcam to use as input for the FER system. Data usage should not be a limiting factor, and it might be possible If used inside a web browser, the back-end server should be computationally strong enough to handle real-time analysis of facial features and their classification. Furthermore, if it is an online application rather than a locally hosted one, it might require a high bandwidth to allow for a high enough framerate to be reached.

The risk of developing this project for a mobile platform is too high, as it would take a lot of effort to develop such a system and getting it to run on a mobile platform, before potentially finding out mobile devices might not be up to the job. Therefore, the program will be run on a personal computer running Windows.

Additionally, at the time of this project, the SARS-CoV-2 pandemic is restricting the possibility of efficient contact with external parties like art galleries or museums,

which are mostly closed and busy adapting to the situation.

In short, the requirements are as follows:

- The intended system should run on a personal computer, preferably with access to a GPU.
- It should run as a standalone application.
- It should not have too big of an impact on the performance of the user's machine.
- It should be testable without personal contact because of the pandemic.

4.3 CONCEPT SELECTION

This section is the transition between this chapter's divergent phase and convergent phase. Based on the information of the two previous sections, the most promising concept is *"Self-reflection tool of behaviour during video games"*. This concept allows for software development based on only little essential feedback during the first prototyping cycle. Furthermore, since it has turned out to be easier to generate various ideas for this concept, more alternatives are available if a prototype happens to hit a dead end in the coming chapters. Finally, this concept is not heavily dependent on expert interviews. Even though they are still valuable sources of information to optimise the workings of the system, replies of experts in this field have more time to provide information, which is an important feat during the pandemic.

From here on, the other projects disregarded and the rest of the design process will focus on this application.

4.4 STAKEHOLDERS

With the proposed tool, i.e. engaging in measurement of an individual's behavioural patterns, many stakeholders will be involved. Stakeholders do not only include the measured individuals, but also experts on relevant subjects in psychology, affective computing, or computer science. Finally, the University of Twente is also involved, monitoring and supporting the ongoing development of this project.

4.4.1 *University of Twente*

The UT is a stakeholder in the sense that this project requires the measurement of individuals that could infringe on their privacy if not done correctly. This supervision on this part will be carried out by the EEMCS ethics committee. Additionally, a requirement for this project is that the technology, i.e. the software, is accessible and clear enough to be used in the curriculum of BSc Creative Technology. This will allow the students to learn Python, while also making themselves acquainted with the field of computer vision and machine learning, among other things.

4.4.2 *Students of BSc Creative Technology*

Since the students will have to familiarise themselves with the code that comes forth from this project, the code will have to be structured efficiently, yet not so efficient that the readability is degraded. This is something the Python programming language is notoriously known for, often referring to the adjective "Pythonic" when something is written compactly and computationally efficiently, sometimes despite the legibility of the code. Plenty of comments and annotations will have to be provided to explain parts of the code that are difficult to read. Finally, although this will not be set as a strict requirement, it would be favourable if all libraries used in the system are cross-platform, or otherwise portable to different operating systems. As the majority of students use Windows-based laptops and (almost) all other people MacOS, this will suffice. Linux and its derivatives will not be taken into account in the scope of this project, although most Python libraries will likely work on it.

4.4.3 *Intended users*

People that are interested for any reason in their behaviour while playing video games are considered a major stakeholder. This is non-specific in terms of any factors like age, gender, or background. Information from this group will be gathered through an online questionnaire, where information is prompted about their individual tendencies to show or experience specific emotions while playing video games, at least to the extent that these people are able to self-report about. For this group, it is important that they are able to gain meaningful insights about their behaviour when using the tool.

4.5 INITIAL CONCEPTUALISATION

With the application and stakeholders identified, the core functionalities of the program can be ideated upon. For each abstract functionality, multiple possible implementations can be thought of in a divergent-convergent style. At the end of this section, the final functionalities should be clear. The technical implementation of these functionalities are discussed in Chapter 5.

Arguably the most intuitive idea is to create a graphical user interface (GUI) that displays relevant information to the user. It should contain a few things:

- An overview of the emotions as have been recorded since the start of the measuring session,
- A view of the user themselves,
- An indication of the current emotion being measured,
- A separate small overlay that is always visible on top of another application.

The overlay could contain various types of information, like current emotion, average emotion, and session duration.

In order to develop this interface efficiently, the following steps are necessary:

1. Create a mock-up of a GUI, which will act as the first prototype capable of receiving feedback.
2. With this feedback, create a first programmatical prototype which is not fully functional yet. The benefit of doing this in this phase is that it provides a great head start for the Specification and Realisation phase. This prototype can be discussed with experts and possible users for more feedback.
3. Implement a functional FER algorithm with an acceptable accuracy into the GUI.
4. Test the finalised prototype GUI with possible users and report their findings, possibly implementing their feedback and iterating this step if time permits.

4.6 FIRST ITERATION

As per the Creative Technology Design method, first a pen-and-paper prototype was created in Adobe Illustrator to get an initial layout set (figure 11). In this design there are a few features for the user to see. The bar chart indicates the confidence levels the FER system per emotion. As the neural network will have a so-called confidence interval for each emotion, the total of which is 100%, this allows the user to see the how confident the system is in predicting their current emotion. The graph below that indicates the history of the user's emotion over time. In essence, it is a history of the bar chart's data. On the bottom right, the relative total of each emotion compared to the other six is displayed. The user also has the ability to view themselves with the found emotion and view some data about the program the user is using, the time and date, the duration of this session, the webcam that is in use, and if the overlay is active or not.

In the top right-hand corner, an overview of the past sessions can be viewed. These sessions can be recorded using the control buttons in the top left-hand corner. This allows the user to enable recording, but also to use the application without having to store the data and using only the real-time data instead. The past session can be compared to each other using the second screen (figure 12). Here, the totals are viewed alongside each other, and the emotion over time can be compared. For example, two sessions of playing the same game can be put against each other.

This prototype, along with a similar-looking programmatical mock-up using the Tkinter Python library were discussed with the supervisor and PhD candidate Johannes Steinrücke, who works on behavioural modelling and is part of the DATA2GAME team in the department of instructional technology at the University of Twente. This department specialises in knowledge acquisition, including cognitive processes of developing and using knowledge and skills, as well as the design of instructional support to facilitate these processes. DATA2GAME investigates "how, and to what extent, the efficacy of computerised training games can be enhanced by tailoring the training scenarios to the individual player." [39]

Between them, the following feedback was provided:

1. Most of the visible data is very exact and precise. This could provoke the user to be very precise themselves and be nit-picky about the accuracy of the measurements.

2. The next iteration should include the aforementioned overlay to see its effects.

3. It is important that the goal of the program is made explicit to the user: does it want you to analyse your own behaviour in-depth and give a little real-time feedback, or does it want you to steer your emotions in real-time, and see an overview of your emotions as backup afterwards. Note that these two cases are each other's opposites.

4. Try to let the interface show mostly the times when there was something interesting going on, and filter out the neutral parts if possible, or make it a user option.

5. If possible, try to incorporate a screen recording when non-neutral emotions are detected.

To process this feedback and implement it into the next iteration, there are a few things to be done. Firstly, the data visualisation should be more intuitive rather than objective and exact. This means that other kinds of visualisations for the same data may provide equally useful insights to the user, possibly more efficiently than the objective data, because it requires less effort to understand. Additionally, the overlay will be part of the next iteration. This iteration will also include the second screen in figure 12, where sessions can be compared to one another.

As it turns out, the implemented classifier is often very confident in whatever prediction it makes. Therefore, the output of the classifier will be a near-binary vector containing only one non-zero entry – (0, 0, 0 ,0, 0, 0, **1.00**) – rather than a vector with two or three non-zero entries – (0, 0, 0, 0, 0, **0.05**, **0.04**, **0.91**) or similar. Although this seems great in principle, showing how strong the algorithm is in detecting the right emotions, it turns out the results don't always match up with what is expected. Going into great technical detail here is meaningless, as there won't be enough time to retrain the classifier. In short, this would be necessary along with a larger amount of training data, or even an alternative neural network architecture, neither of which are within the scope of this thesis. What is important is its implications for the data visualisation, namely that a binary time series (a line graph with only zeroes and ones) would be an odd choice. Therefore, in the next iteration a different visualisation will have to be found.

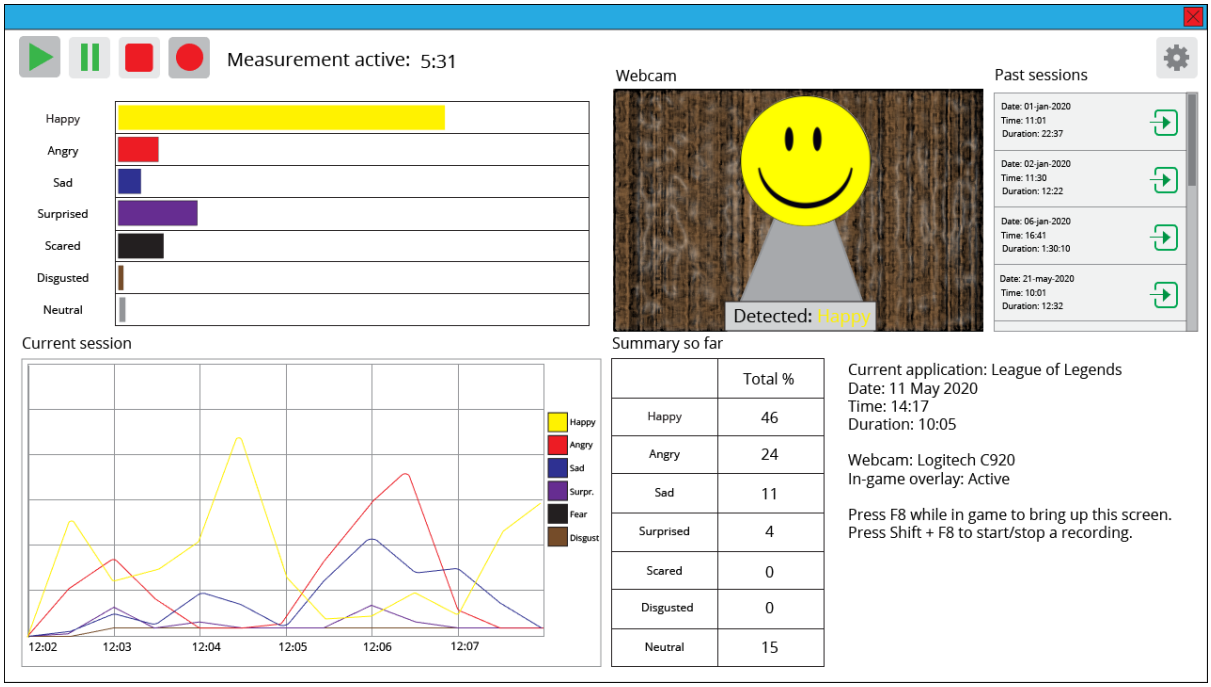

Figure 11: The main dashboard of this system.

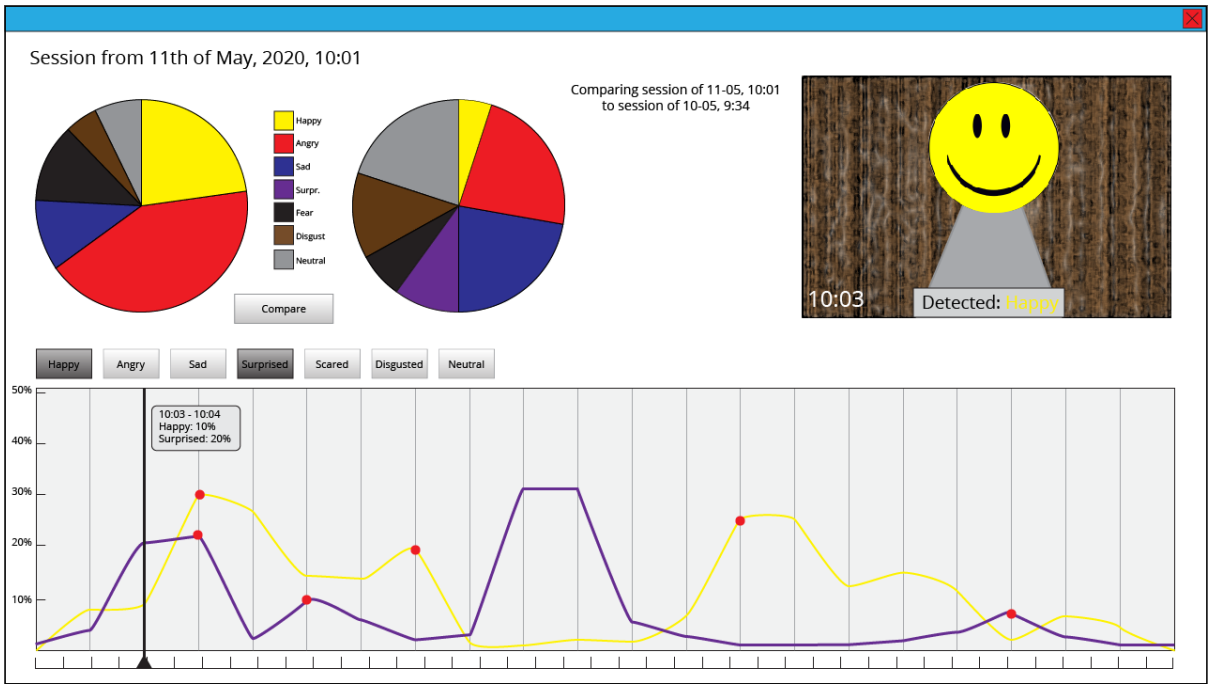

Figure 12: The comparative overview of two different sessions.

5

SPECIFICATION

In this chapter, the goal is to improve on the first prototype by "further refining and prioritising the user requirements and obtain the best possible functional specification of the envisioned product prototype" [38]. The end result of this phase is to have a clear overview in which all requirements, needs, possible features, and nice-to-haves are enumerated, which will be based on user experience tests and the results of the first prototype.

However, not all ideas and suggestions can be turned into reality due the timely nature of graduation projects. A commonly used tool to separate and discovery importance is the MoSCoW method. This creates a list which separates its content into four categories: "Must have", "Should have", "Could have", and "Won't have". Here, "Must have" entries are essential for the correct working of the product, and the endeavour should be considered a failure if these needs aren't met. "Should have" entries are important but not absolutely necessary for delivery within the current timespan. "Could have" are ideas that would benefit the final product and the user experience that comes forth from it, but are usually only included if there is enough time. Finally "Won't have" entries are the least-critical additions or elements. Any ideas here are not discarded, but also not used for a next iteration in a new timespan. It is good to include these nevertheless to indicate that the feature or idea has been considered). The suggestions provided in Chapter 4 are categorised in the table below.

With the added requested functionalities organised, it is possible to make a complete overview of all the user needs so far:

The user needs to be able to...

1. Record themselves to see how their facially expressed behaviour changes over time.
2. Store and review a session at a later time, with context in the form of screenshots.
3. Be able to compare different sessions with each other.
4. See their emotions in an unobtrusive way while the session is ongoing.
5. Understand all the data that is visualised without a lot of explanation.

In order to achieve these goals, the proposed changes should be clarified. The on-screen overlay will consist of a small, unobtrusive text box that sits on top of the application you're running. In this text box, the emotion that the system is measuring through your webcam is displayed in real-time. This allows the user to not only reflect after stopping the session, but also during that session. If a user notices that they are angry a lot, they might take attempt to change their behaviour on the fly. If speed and efficiency permits, a small preview of the webcam could accompany the text box, so

that the user knows what the system is "seeing" and can determine how accurate it is. This is important since the classifier is probably not one hundred percent accurate, and might be unintentionally biased towards a specific outcome based on lighting, skin colour, camera angles, etc.

The data visualisation will be changed from a line plot-style time series to a word cloud that contains the emotions. This means that he user can get a quick overview of the total of their emotions during a session from the system's dashboard, without having to process all the numerical data. This enforces the usage of the dashboard on a second monitor, for example, as this data is perfectly suited for being read in a short time, which is ideal when playing video games.

Session comparison will not be an explicit feature. However, each session can be opened in a separate window with an identical layout, so it should still not be too difficult to compare the results. In the session comparison screen, a timeline is constructed from all the emotive data points. If technically feasible, each data point could be accompanied by a screenshot of the monitor the user is playing on. If a data entry is clicked, the appropriate screenshot can be displayed to give the user context as to what happened. An alternative idea is to continuous screen recording to allow for constant playback as well as audio playback, but because of the expected strain of the system, screenshots are opted for instead. This will likely cause a large amount of disk space to be required prior to starting each session, especially in the images are stored without compression.

Finally, instead of filtering out the "Neutral" emotions, the timeline in the session screen will be using an event plot. This plot contains seven synchronised timelines, one for each emotion. At the point where a given emotion is detected, a coloured bar is placed. These timelines the combine to a total timeline of all emotive data, which can be filtered by emotion. If the user wants to see only non-neutral data, they can opt for that in this way instead. This is both a better way to represent the available data, as well as a less intensive overhaul to the first prototype.

Regarding the implementation of the FER system, it is unlikely that it is possible to design, train and implement a custom classification system. Therefore, an open-source neural network will be used. Neural networks are the most common because many people have interests in them and they are relatively easy to create, so there should be more options available than more intricate or elaborated classification systems. This will likely be at the cost of detection accuracy. In turn, this will likely impact any evaluation by test users, but in the grander scope of things it is not a big problem, since the network will be implemented modularly, and should be easy to replace with a better one in a next GP using the source code from this project.

6

REALISATION

In this chapter, the final version of the prototype is discussed, improving on the first prototype discussed in Chapter 4, with the improvements and goals from Chapter 5. This is the final phase of the main design process; no further modifications will be made to the code after this chapter.

To fully understand the workings of the system, and because it is good coding practice in general, each high-level functionality is contained within its own module. Below all modules are explained, sometimes with the help of pseudo-code. The full code repository can be found on GitHub.

6.1 GRAPHICAL USER INTERFACE

The final GUI consists of two window classes and one overlay class. The first window class, or dashboard, is in many ways the same as the "pen-and-paper" design from before. The GUI is mostly written using the `Tkinter` library. This comes pre-installed with Python 3 and is easy to set up. The initial layout was created by using PAGE [40]. This made it easy to get a first sense of the layout. The graphs are made using `matplotlib` which can be drawn into `Tkinter` windows. All GUI related classes are found in the `gui.py` module. This makes it easier to communicate between the separate windows.

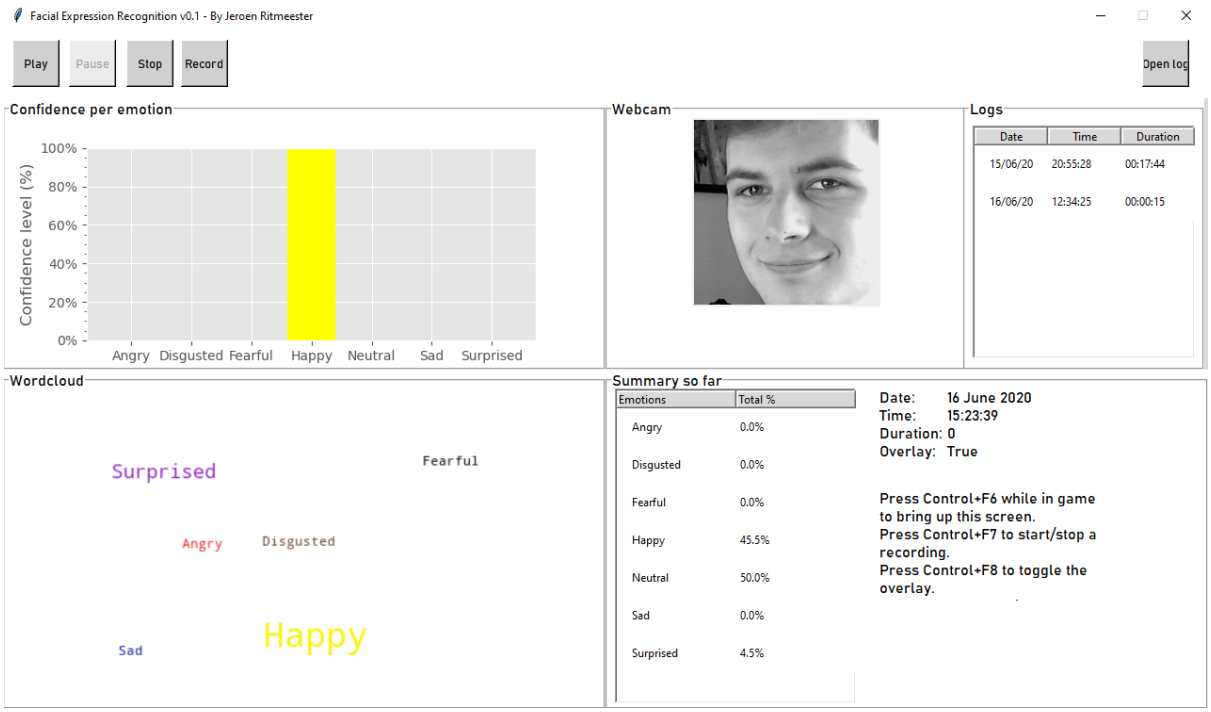

Figure 13: Final dashboard design

6.2 DASHBOARD

In Figure 13 the dashboard is visible after it has been running for approximately ten seconds, and was paused after that. The bar chart indicates a one hundred percent confidence in the prediction "Happy", and the webcam frame it detected it on is shown on the side. This is a cropped and black and white view on purpose, because this is the frame that the neural network bases its decisions on. If the prediction is wrong, the user help improving it by seeing if there is anything weird in the frame. For example, one side of the face might be facing a window, causing it a part of the face to be completely white, which reduces the reliability of the prediction. More information on the workings of the neural network can be found in Section 6.3.

On the bottom left there is the word cloud containing the emotions, with their size being an indicator of their relative frequency. As discussed earlier, this is very helpful for high-concentration games or other applications, that do not allow for in-depth analysis of live data. Instead, this gives a quick overview of what has happened so far. To introduce slightly more detail, though, the table to the right of it contains all the relative frequencies expressed in percentages.

On the far right side, the completed logs are shown with their date, start time, and duration are shown. This view is updated live every time the user presses 'Stop', so that the user can access the latest sessions. Finally, in the bottom right, some live updated details and hotkey commands are explained. These hotkeys were chosen in the assumption that few other applications use these combinations, in order to avoid interference.

6.3 SESSION VIEW

In the session view, the user can gain more insight into the emotions they have showed during the recording session. The `SecondWindow` class contains a pie chart with the totals of the measured emotions on the top-left, a screenshot for context on the top-right, and the timeline at the bottom. This `eventplot` shows the timeline as was discussed in the previous chapter. Using the buttons in the centre of the screen, any emotion can be filtered out. The plot is interactive, so the user has the ability to move the plot around, scale it, and even save an image of the plot to their computers for later reference.

If any data point in the plot is clicked, the x value of the point in the timeline is used to calculate the data point closest to that position. Since each data point has an accompanying image, this makes it possible to also find the image to show in the top-right. To do this, the distance to all data points is calculated by taking their squared differences, and returning the data point with the index equal to the index of the lowest difference found:

```
for each data_point in all_data_points:
    distance = (data_point.time - mouse_x)**2
nearest_data_point = data_points[argmin(distance)]
```

Here, `argmin` returns the index of the lowest distance.

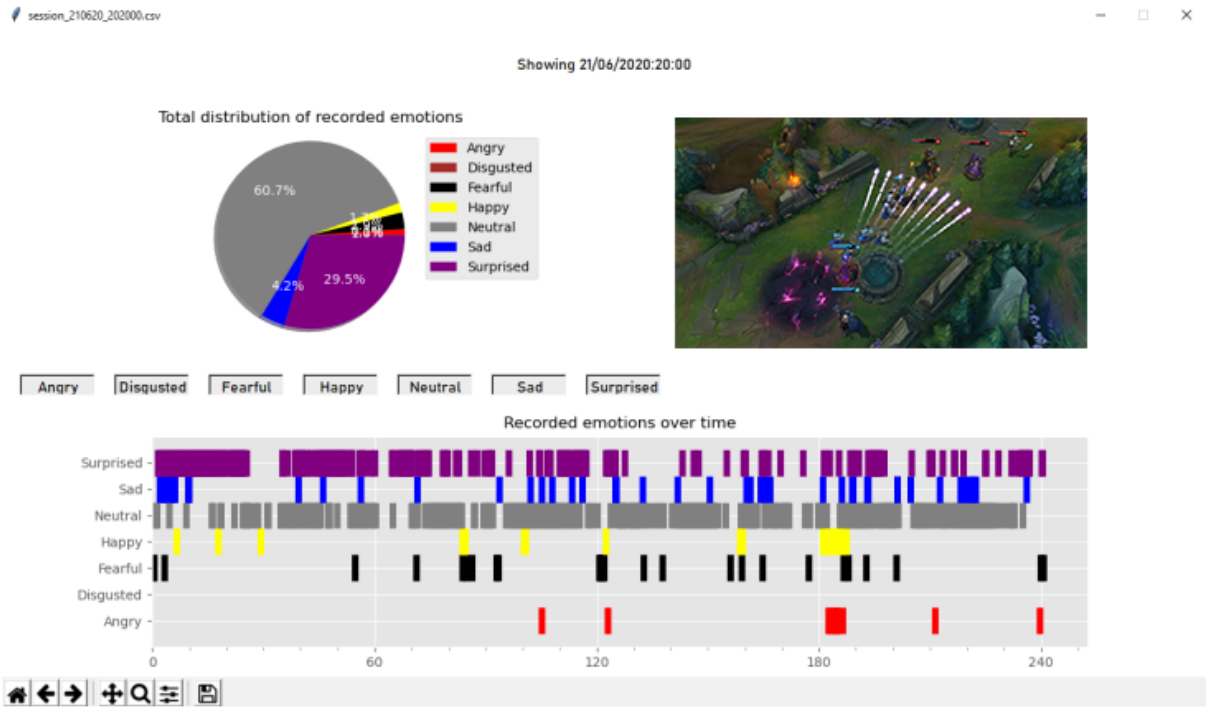

Figure 14: Final session view making use of the eventplot and in-game screenshots

6.4 EMOTION DETECTION

Most computer program try to run at the highest speed possible unless limited. Although that sounds great in theory, instead it will produce a lot of high-definition data that server no extra use and uses a lot of resources. Therefore the system is limited to measure the facial expressions once every second. This means that expressions are only missed if they last shorter than one full second and occur precisely in between two measurements.

The module `emotions.py` contains the emotion tracking code. Before the FER starts, the CNN in built from various parts (see next section), and the pre-trained weights are loaded. Since the program runs indefinitely, it extends the `Thread` class, which means the GUI and the FER system can work at the same time, i.e. they don't have to wait for each other to finish processes. In this thread, the webcam is read once every second, the face is detected using Haar-like features using `OpenCV's CascadeClassifier`, and the face in grayscale is displayed in the GUI. Finally, the input image is then downscaled to a one-channel (grayscale), 48x48 image and fed into the neural network for emotion detection. From here, the current emotion, each emotion's total times it has been detected, and raw prediction data is stored.

6.5 CONVOLUTIONAL NEURAL NETWORK

The Facial Expression Recognition system revolves around a convolutional neural network by Atul Balaji [41]. Although the network is not particularly strong, it was already pre-trained on the FER2013 image data set, which is widely used in the field of computer vision. Knowing that something is trained on a popular data set reduces

the likelihood of low accuracy caused by poor image variability or low data availability. Note that the data set does not include the emotion contempt.

The data set consists of two parts: a training set and a test set. The training set consists of 28,709 images, each 48x48 pixels, all black-and-white (grayscale). These images are fed into the network, accompanied by the a number between 0 and 6 (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). Using the predictions and the known outcomes, the network tries to get as good as possible at getting the right prediction, which is tested with the test set, consisting of 3,589 unique images that were not present in the training set.

This neural network design makes use of 2D convolutions for feature detection. This means the input images (a person's face) is processed by summing pixels in groups according to the kernel size. This can be done any number of times to a three dimensional object containing specific features. What these features are, and more importantly what they represent, is determined by the adjustment of the weights between each layer. One filter could be picking up on high contrast areas like the eye sockets or nose, another could be finding specific shapes like curves or corners. This process is repeated multiple times in slightly different ways. For example, the next operation could have a different kernel size, depth (amount of filters per operation), et cetera.

In between operations, a dropout operation may be introduced, which removes some connections or weights, thus introducing a slight instability in the system. This forces the system to find more ways to accomplish its goal. This prevents overfitting, which means that the system is too focused on some features, while ignoring others, often leading to bad result on images that are not like the data set it is trained on.

Finally, the dimensionality is reduced to one: all values are turned into a long string of numbers, which is turned into a string of numbers that is 7 long. There, each number will indicate a confidence interval between 0 and 1.0, and the most confident output (highest number) is the most likely detected emotion.

6.6 SCREENSHOTS

The screenshots that provide the in-game context that is used in the session view are taken every time a data point is logged. As the system as a whole is not timed, for example by using the `Threading` module, the system will try to run as fast as possible in order to record as many data points as possible. This is mostly to approximate a continuous measurement, since emotions are unpredictable, as are the event that trigger them. At first, the `MSS` module was used to gather uncompressed `.png` screenshots of the users primary monitor. As most people nowadays run a screen of dimensions 1920x1080 pixels, this uncompressed format took up over five gigabytes of disk space within twenty minutes of logging. Therefore, the system now makes use of `PIL` that can take `.jpg` format screenshot, allowing for a much better efficiency regarding disk space. As mentioned earlier, the filenames correspond to the relative timestamp of the system, allowing for easy look-up in the session overview's `eventplot`.

6.7 OVERLAY

The overlay is a small, 100x40 pixel `Tkinter` window with the title bar removed. It is located at the top-left of the screen, but lowered 100 pixels from the top. This is so that any in-game GUI options, which are often located on the edges of the screen, are not obscured. The window is also forced to sit on top of any other windows that are active. This means that is visible when playing games, as long as the game is not set to fullscreen mode. The overlay can be toggled on and off according to the user's preference using the `Control+F8` key combination.

The overlay gets updated every time the FER system detects a new face and the emotion is updated. Each update, one of six white background images are displayed, each containing the emotion as a word in their respective colours. This was found to be more reliable than using `Tkinter`'s `canvas` feature, or drawing text straight to the screen.

In a prototype in between the ideation phase's first prototype and the one discussed here, an option was added to duplicate the webcam preview shown in figure 13 just above the text, so that the user could judge in real-time how the system was performing accuracy-wise. However, this required more rescaling of the preview to fit nicely, which ended up requiring more computational power, and was found to be very distracting during gameplay. Therefore it was removed in the final prototype.

6.8 ACTIVITY DIAGRAMS

Depicted in figures 15 and 16 are the two ways the system can be utilised. In the former, live mode is used. This means that no data is being logged while the FER system is active. When the user stops the system, all data is discarded. The primary function of this is quick usage of the system when data storage or later reference is not required. In the latter, the data is logged into a `.csv` file, along with the screenshots that accompany each data point.

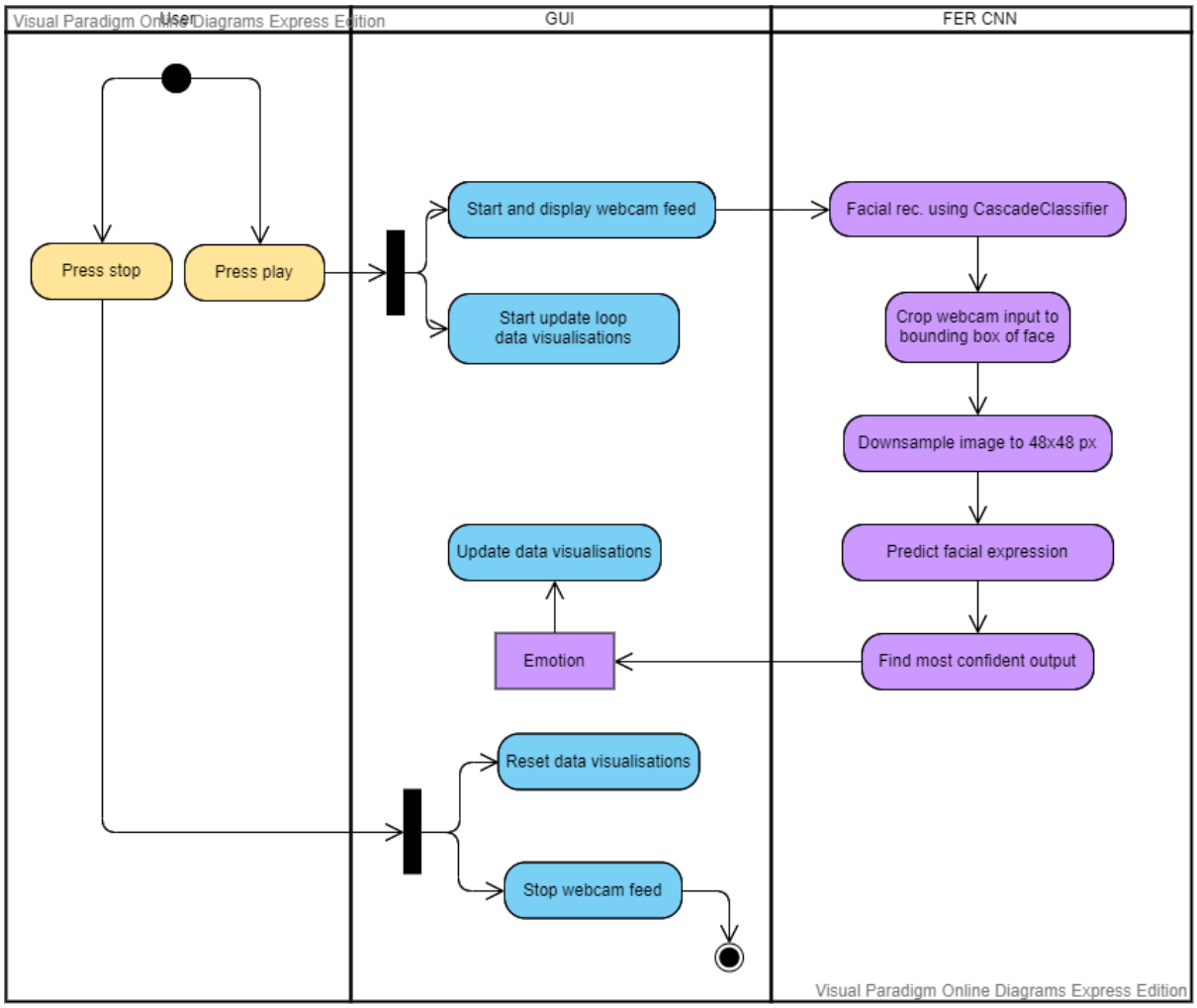

Figure 15: Activity diagram of using live mode

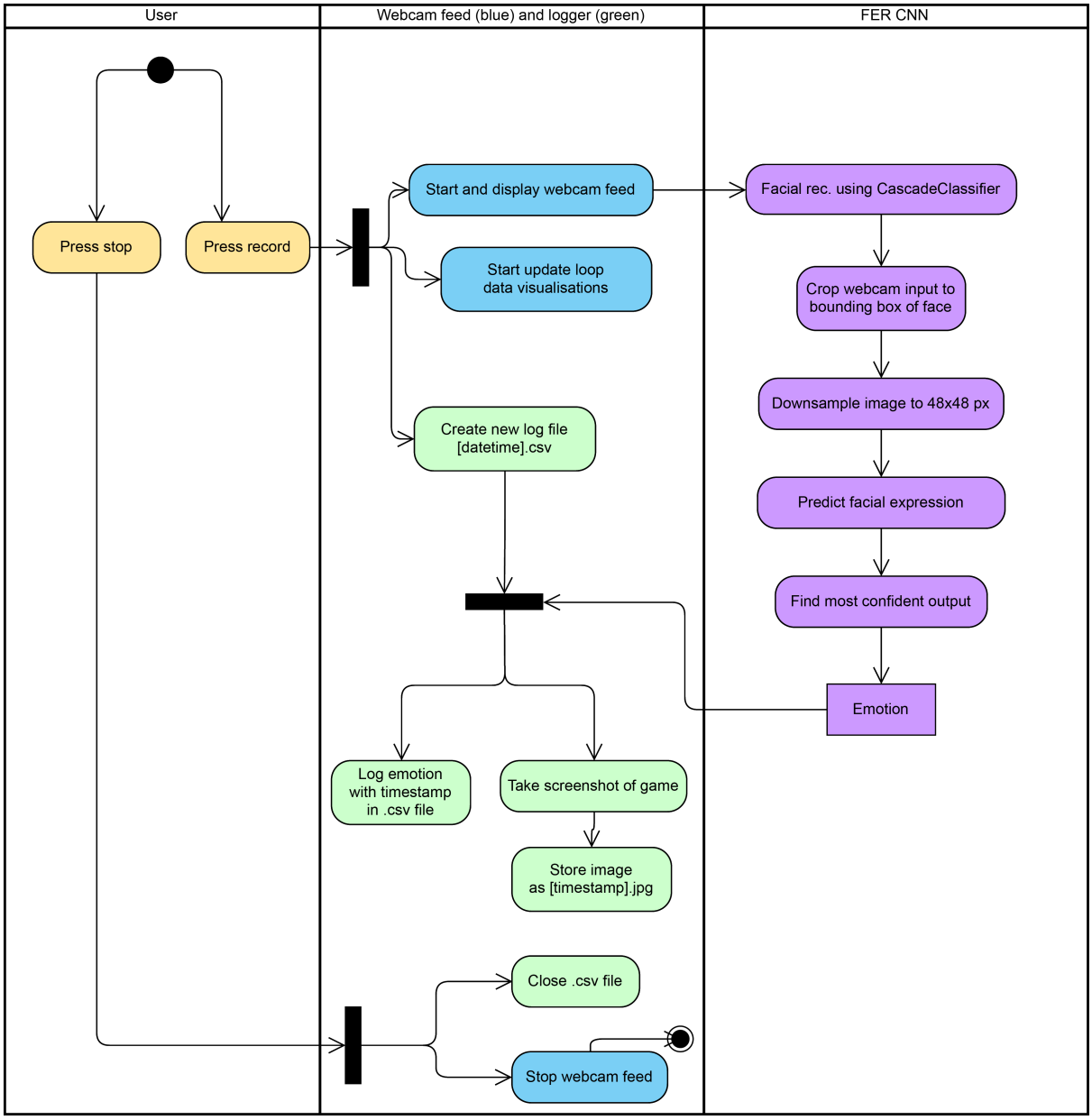

Figure 16: Activity diagram of using recording mode

7 EVALUATION

As this system is focused on the user's appreciation for a system that uses FER for emotional self-reflection, the evaluation focuses largely on non-numerical, subjective data. Users opinions were summarised and are reported here to get an idea of how well the goal of this thesis has been reached, as well as the functionalities of the intended system.

The system was evaluated with the test users in their own home to avoid any potential health risks caused by the ongoing pandemic. The test user group consists of three men and three women, all around the age of twenty. Each user installed the program through `Github` along with a instruction manual that showed them how to install `Python 3.8` properly, install the system using `pip` and operate the prototype. Over the course of a few days, they were all asked to interact with the system to their liking for a total minimum of one hour, whilst playing a video game of their liking. After that week, they were each interviewed for approximately 20 minutes. The questions can be found in Appendix A.

7.1 GENERAL EXPERIENCE

Overall, the user group enjoyed using the system, as it showed them a unique insight into their behaviours. Each user reported that it was interesting to discover recurrent patterns in their behaviour throughout their gaming sessions, like getting angry at a particular event, or looking scared without them realising. Out of ten, the average satisfaction grade - an indicator of the frustration or enjoyment caused by using the prototype - is a 6,7. Some comments indicate that the joyful experiences during a game were enhanced by looking back at the moment in happened, and seeing back the emotional recording. Others indicated that the recognition system was easy to influence, or overall inaccurate. Luckily, the system did not cause a drastic performance drop.

Unfortunately, almost all users reported the lack of recurring interest in using the system. When asked for an explanation, their answers were similar too: none of them had any behavioural issues during gaming that they would like to rectify or adjust. Because of this, apart from anything out of the ordinary would occur in the game, no emotional patterns could be shown that would activate the user to self-reflect on that moment and their behaviour. One user suggested that this system would be interesting to them when used outside the scope of gaming, like browsing the internet or working. This is good news for the future of this system, as many more groups of people can be included in future tests.

When asked what they learned about themselves, the answers were all very similar as well. As time went on in their gaming session, according to the users, the surprise emotion decreased and made way for anger, which was often already prevalent. Be-

cause of the limited time span, the users logs and its contents were not analysed in detail, so no distribution of emotion detection can be constructed.

7.2 EMOTION DETECTION

The system's accuracy is not a part of the scope of this system. However, a lack of accuracy could influence the user's likelihood to utilise the system in the future. Naturally, if the prototype lacks any accuracy at all, users can get frustrated. For that reason, they were also asked to grade the accuracy, to get a sense of what the accuracy is like to them. This grade averaged on a 4,8, which can be considered insufficient. Shown below is a screenshot of one of a user's logs. In the timeline, two distinct halves can be made out: in the first half, the user was wearing glasses, specifically with thick frames. In the second part, the user took off their glasses. There, apart from the bias towards detecting fear, the classifier performs significantly better.

One user suggested that they were far more likely to use the system more often in the future if the accuracy is increased to a sufficient level. For ideas on how to further improve the accuracy, see Chapter 8.

7.3 GRAPHICAL USER INTERFACE

The interface was considered clean and informative. The users had not been given any explanation of the data visualisations or what they represented, but during the interviews all of them could provide a correct explanation for each of their purposes. The users gave the interface an average grade of an 8,3. The main dashboard received only two practical negative remark, namely that the webcam should be activated and shown before pressing the play button in order to prevent thinking that the webcam is not working, and that the session log list should be made somewhat clearer. Some users were confused by the fact that a log first had to be clicked and then opened using a separate button.

The session view received a lot of praise regarding clarity as well. Some confusion over the pie chart showing the total emotions was discussed; this is mostly due to the way it also includes neutral data points. Apart from that, all features worked as intended, with no crashes or bugs reported. The overlay was well-received too, especially in conjunction with the fact that it can be turned on or off, so that any in-game GUI that occupies the same part of the screen can be accessed. No improvements were suggested for the overlay.

Each user was also asked if there was a statistic, data visualisation or something similar, that they would like to see in a future iteration. Four suggestions were made:

- The face snapshot used in the pen-and-paper prototype should exist alongside the in-game screenshots in the session overview.
- A sound level indicator could also be present as indicator of arousal. Alternatively, the screenshots and audio could of course be combined to a continuous video playback.

- The user should be able to select which camera is being utilised in case more than one video capture device is connected to the user's system.
- A hotkey can be made for the insertion of a time marker in the `eventplot`, so that the users can mark a point during the recording that they want to find later in the session view.

8

CONCLUSIONS AND DISCUSSION

8.1 CONCLUSIONS

This graduation project has been an attempt at developing a way of introducing both potential users as well as potential developers to FER. After having created a fully functional prototype, it is now possible to see to what extent this goal has been reached by answering the earlier posed research questions, which were the following:

- How can a tool be designed to provide emotive self-reflection to people playing video games?
- How can that tool be implemented so that it raises awareness of the technology's potential?
- To what extent is there a place for such a tool among video gamers?

It would now appear that the first two sub-questions share the same answer; creating this prototype and exposing it to as many people as possible, both for casual use or educational use, raises the awareness of Facial Expression Recognition, as it is the only noteworthy technology that comes into play. For finer details of the first sub-question, this entire design process is an account of its development.

The last sub-question can be answered only partially for a number of reasons. The first is that the test user group was smaller than required to get a more complete image of the desire of the gaming community for a tool like presented here. The second reason is that the users that did participate indicated that they did not have any behavioural issues or negative patterns that they would like to adjust. It is now clear that in order for such a tool to be used properly, the user needs an incentive from themselves - in which case they are self-aware of their emotions - or outside, like family, social circles, or a psychologist. To accommodate this requirement in the future, tests should be used with much larger user groups, consisting of users that have such an incentive to use the software more intensively.

Finally then, this leads to an answer of the main design question:

- How can the facial expression recognition be integrated into a system that allows creating awareness of potential of this technology?

Over the course of six months, an emotive self-reflection tool making use of Facial Expression Recognition has been developed. Making use of a convolutional neural network, six of the seven emotions presented in the basic emotion theory can be used to create a session-based recording system that allows the user to review their own emotions, and by extension, their behaviour. By proper documentation and a clear, mostly self-explanatory graphical user interface, this tool allows gamers and developers to learn about the potential that FER has to offer.

8.2 DISCUSSION AND FUTURE WORK

Over the course of this project, it became clear that a number of parts of the research process could be improved upon. First and foremost, the evaluation of the final prototype could have been much more concrete if a way was devised to allow for numerical and more systemic feedback. Although for the first major prototype it is not essential as many large changes may be made in the next iteration, some statistical data could have been the basis of a more well-founded argument for or against choices that have been made.

As mentioned in the previous section, it would be a good idea to put more time into forming a larger, more specific test user group, or possibly even multiple specific test user groups, each for a different application (gaming, work from home, studying, et cetera).

Finally, the upgrade to a better CNN is crucial for this project's future. Without a proper classifier, this project cannot be evaluated sufficiently. It is a good idea to either create a custom classifier, or search for one with a higher accuracy. In fact, different types of classifier may prove to be a valuable asset. As became apparent during the pandemic, where online video calls had to substitute for face-to-face meetings, a lot of emotion is lost in transmission. If an active learning classifier is to be used that is trained on a large data set, but improves over time with usage by reading a specific user's face, the system can improve its performance and personalisation. This means that it is possible for any biases towards specific emotions - which were abundant during this project - to be reduced into insignificance. This would still be in line with the goals set out in this project, likely being more effective even at reaching more people with the FER.

BIBLIOGRAPHY

[1] Christina Gough. *Number of gamers worldwide 2021*. 2019. URL: https://www.statista.com/statistics/748044/number-video-gamers-world/ (visited on 25/05/2020).

[2] *Newzoo's Global Games Market Report | Newzoo Platform*. 6th July 2020. URL: https://newzoo.com/products/reports/global-games-market-report/ (visited on 06/07/2020).

[3] Paul Ekman, Wallace V. Friesen and Sonia Ancoli. 'Facial signs of emotional experience'. In: *Journal of Personality and Social Psychology* 39.6 (1980), pp. 1125–1134. ISSN: 00223514. DOI: 10.1037/h0077722.

[4] Mohammad H. Mahoor et al. 'A framework for automated measurement of the intensity of non-posed facial action units'. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*. 2009. ISBN: 9781424439911. DOI: 10.1109/CVPR.2009.5204259.

[5] Paul Ekman and Wallace V Friesen. 'EMFACS-7'. In: (1983).

[6] Paul Ekman. *Facial Action Coding System (FACS) - A Visual Guidebook*. URL: https://imotions.com/blog/facial-action-coding-system/%7B%5C#%7Dmain-action-units (visited on 17/06/2020).

[7] Marian Stewart Bartlett et al. 'Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction.' In: *2003 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE, June 2003, pp. 53–53. ISBN: 0769519008. DOI: 10.1109/CVPRW.2003.10057. URL: http://ieeexplore.ieee.org/document/4624313/.

[8] Marc Archinard, Véronique Haynal-Reymond and Michel Heller. *Doctor's and patients' facial expressions and suicide reattempt risk assessment*. 2000. DOI: 10.1016/S0022-3956(00)00011-X.

[9] Kenneth M. Prkachin and Patricia E. Solomon. 'The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain'. In: *Pain* (2008). ISSN: 03043959. DOI: 10.1016/j.pain.2008.04.010.

[10] Zhanli Chen, Rashid Ansari and Diana J. Wilkie. 'Automated detection of pain from facial expressions: a rule-based approach using AAM'. In: *Medical Imaging 2012: Image Processing*. 2012. ISBN: 9780819489630. DOI: 10.1117/12.912537.

[11] Gwen C. Littlewort, Marian Stewart Bartlett and Kang Lee. 'Faces of pain: Automated measurement of spontaneous facial expressions of genuine and posed pain'. In: *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI'07*. 2007. ISBN: 9781595938176. DOI: 10.1145/1322192.1322198.

[12] Jeffrey M. Girard et al. 'Spontaneous facial expression in unscripted social interactions can be measured automatically'. In: *Behavior Research Methods* (2014). ISSN: 15543528. DOI: 10.3758/s13428-014-0536-1.

[13] Paul Ekman. *Facial Action Coding System - Paul Ekman Group*. URL: https://www.paulekman.com/facial-action-coding-system/ (visited on 09/04/2020).

[14] Takeo Kanade, Jeffrey F. Cohn and Yingli Tian. 'Comprehensive database for facial expression analysis'. In: *Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000*. 2000. ISBN: 0769505805. DOI: 10.1109/AFGR.2000.840611.

[15] Patrick Lucey et al. *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression*. \url{https://ieeexplore.ieee.org/document/5543 June 2010. DOI: 10.1109/CVPRW.2010.5543262. URL: http://ieeexplore.ieee.org/document/5543262/.

[16] Elżbieta Kukla and Paweł Nowak. 'Facial emotion recognition based on cascade of neural networks'. In: *Advances in Intelligent Systems and Computing* 314 (2015), pp. 67–78. ISSN: 21945357. DOI: 10.1007/978-3-319-10383-9_7.

[17] Daniel McDuff and Rana El Kaliouby. 'Applications of Automated Facial Coding in Media Measurement'. In: *IEEE Transactions on Affective Computing* 8.2 (Apr. 2017), pp. 148–160. ISSN: 19493045. DOI: 10.1109/TAFFC.2016.2571284.

[18] Sina Shafaei, Tahir Hacizade and Alois Knoll. 'Integration of Driver Behavior into Emotion Recognition Systems: A Preliminary Study on Steering Wheel and Vehicle Acceleration'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11367 LNCS. Springer Verlag, Dec. 2019, pp. 386–401. ISBN: 9783030210731. DOI: 10.1007/978-3-030-21074-8_32.

[19] J. F. Cohn et al. 'Individual differences in facial expression: Stability over time, relation to self-reported emotion, and ability to inform person identification'. In: *Proceedings - 4th IEEE International Conference on Multimodal Interfaces, ICMI 2002*. 2002. ISBN: 0769518346. DOI: 10.1109/ICMI.2002.1167045.

[20] Rosalind W. Picard. *Affective Computing*. 1997. ISBN: 0262161702. DOI: 10.5555/265013.

[21] Jonathan Posner, James A. Russell and Bradley S. Peterson. 'The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology'. In: *Development and Psychopathology* (2005). ISSN: 09545794. DOI: 10.1017/S0954579405050340.

[22] James A. Russell. 'Core Affect and the Psychological Construction of Emotion'. In: *Psychological Review* (2003). ISSN: 0033295X. DOI: 10.1037/0033-295X.110.1.145.

[23] Charles Darwin. 'The Expression of the Emotions in Man and Animals.' In: *The Journal of the Anthropological Institute of Great Britain and Ireland* (1873). ISSN: 09595295. DOI: 10.2307/2841467.

[24] David L. Robinson. 'Brain function, emotional experience and personality'. In: *Netherlands Journal of Psychology* (2008). ISSN: 1872-552X. DOI: 10.1007/bf03076418.

[25] Titus Thomas, Miguel Dominguez and Raymond Ptucha. 'Deep independent audio-visual affect analysis'. In: *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, Nov. 2017, pp. 1417–1421. ISBN: 978-1-5090-5990-4. DOI: 10.1109/GlobalSIP.2017.8309195. URL: http://ieeexplore.ieee.org/document/8309195/.

[26] Liang Chih Yu et al. 'Building Chinese affective resources in valence-arousal dimensions'. In: *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*. 2016. ISBN: 9781941643914. DOI: 10.18653/v1/n16-1066.

[27] Yading Song et al. 'Do online social tags predict perceived or induced emotional responses to music?' In: *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013*. 2013. ISBN: 9780615900650.

[28] Jeroen Jansz. *The emotional appeal of violent video games for adolescent males*. 2005. DOI: 10.1093/ct/15.3.219.

[29] Craig A. Anderson and Brad J. Bushman. 'Effects of violent video games on aggressive behavior, aggressive cognition, aggressive affect, physiological arousal, and prosocial behavior: A Meta-Analytic Review of the Scientific Literature'. In: *Psychological Science* (2001). ISSN: 09567976. DOI: 10.1111/1467-9280.00366.

[30] Isabela Granic, Adam Lobel and Rutger C.M.E. Engels. 'The benefits of playing video games'. In: *American Psychologist* (2014). ISSN: 0003066X. DOI: 10.1037/a0034857.

[31] Daphne Bavelier et al. *Brains on video games*. 2011. DOI: 10.1038/nrn3135.

[32] Sicheng Zhao, Hongxun Yao and Xiaoshuai Sun. 'Video classification and recommendation based on affective analysis of viewers'. In: *Neurocomputing* 119 (2013), pp. 101–110. ISSN: 09252312. DOI: 10.1016/j.neucom.2012.04.042.

[33] T. F. Cootes et al. 'Active shape models - their training and application'. In: *Computer Vision and Image Understanding* (1995). ISSN: 10773142. DOI: 10.1006/cviu.1995.1004.

[34] Peng Yang, Qingshan Liu and Dimitris N. Metaxas. 'Exploring facial expressions with compositional features'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010. ISBN: 9781424469840. DOI: 10.1109/CVPR.2010.5539978.

[35] Peng Yang, Qingshan Liu and Dimitris N. Metaxas. 'Boosting encoded dynamic features for facial expression recognition'. In: *Pattern Recognition Letters* (2009). ISSN: 01678655. DOI: 10.1016/j.patrec.2008.03.014.

[36] Kushsairy Kadir et al. 'A comparative study between LBP and Haar-like features for Face Detection using OpenCV'. In: *2014 4th International Conference on Engineering Technology and Technopreneuship, ICE2T 2014*. 2015. ISBN: 9781479946211. DOI: 10.1109/ICE2T.2014.7006273.

[37] Hideo Joho et al. 'Looking at the viewer: analysing facial activity to detect personal highlights of multimedia contents'. In: *Multimedia Tools and Applications* 51.2 (Jan. 2011), pp. 505–523. ISSN: 1380-7501. DOI: 10.1007/s11042-010-0632-x. URL: http://link.springer.com/10.1007/s11042-010-0632-x.

[38] Angelika Mader and Wouter Eggink. 'A design process for Creative Technology'. In: *Proceedings of the 16th International Conference on Engineering and Product Design Education: Design Education and Human Technology Relations, E and PDE 2014*. 2014. ISBN: 9781904670568.

[39] *Home | DATA2GAME*. 11th May 2020. URL: https://www.data2game.nl/ (visited on 06/07/2020).

[40] *PAGE - A Python GUI Generator*. 29th June 2020. URL: http://page.sourceforge.net/ (visited on 06/07/2020).

[41] Atul Balaji. *Real-time Facial Emotion Detection using deep learning*. 2017. URL: https://github.com/atulapra/Emotion-detection (visited on 16/06/2020).